FLEX 10KE

Embedded Programmable Logic Family

May 1999, ver. 2 Data Sheet

Features...

Preliminary Information

- Embedded programmable logic devices (PLDs), providing System-on-a-Programmable-Chip$^{\text{TM}}$ integration in a single device
  - Enhanced embedded array for implementing megafunctions such as efficient memory and specialized logic functions
  - Dual-port capability with up to 16-bit width per embedded array block (EAB)
  - Logic array for general logic functions
- High density
  - 30,000 to 200,000 typical gates (see Tables 1 and 2)
  - Up to 98,304 RAM bits; 4,096 bits per EAB, all of which can be used without reducing logic capacity
- System-level features
  - MultiVolt$^{\text{TM}}$ I/O pins can drive or be driven by 2.5-V, 3.3-V, or 5.0-V devices
  - Low power consumption
  - Bidirectional I/O performance (t_{SU} and t_{CO}) up to 250 MHz
  - Fully compliant with the peripheral component interconnect Special Interest Group (PCI SIG) ***PCI Local Bus Specification, Revision 2.2*** for 3.3-V operation at 33 MHz or 66 MHz
  - -1 speed grade devices are compliant with ***PCI Local Bus Specification, Revision 2.2***, for 5.0-V operation
  - Built-in Joint Test Action Group (JTAG) boundary-scan test (BST) circuitry compliant with IEEE Std. 1149.1-1990, available without consuming additional device logic

For information on 5.0-V FLEX 10K or 3.3-V FLEX 10KA devices, see the *FLEX 10K Embedded Programmable Logic Family Data Sheet* in this data book.

Table 1. FLEX 10KE Device Features

Feature	EPF10K30E	EPF10K50E EPF10K50S	EPF10K100β
Typical gates *(1)*	30,000	50,000	100,000
Maximum system gates	119,000	199,000	158,000
Logic elements (LEs)	1,728	2,880	4,992
EABs	6	10	12
Total RAM bits	24,576	40,960	24,576
Maximum user I/O pins	220	254	191

Table 2. FLEX 10KE Device Features

Feature	EPF10K100E	EPF10K130E	EPF10K200E EPF10K200S
Typical gates *(1)*	100,000	130,000	200,000
Maximum system gates	257,000	342,000	513,000
Logic elements (LEs)	4,992	6,656	9,984
EABs	12	16	24
Total RAM bits	49,152	65,536	98,304
Maximum user I/O pins	338	413	470

Note to tables:

(1) The embedded IEEE Std. 1149.1 JTAG circuitry adds up to 31,250 gates in addition to the listed typical or maximum system gates.

...and More Features

- Fabricated on an advanced process and operate with a 2.5-V internal supply voltage
- In-circuit reconfigurability (ICR) via external configuration devices, intelligent controller, or JTAG port
- ClockLock™ and ClockBoost™ options for reduced clock delay/skew and clock multiplication
- Built-in low-skew clock distribution trees
- 100% functional testing of all devices; test vectors or scan chains are not required
- Pull-up on I/O pins before and during configuration

▩ Flexible interconnect
  - FastTrack® Interconnect continuous routing structure for fast, predictable interconnect delays
  - Dedicated carry chain that implements arithmetic functions such as fast adders, counters, and comparators (automatically used by software tools and megafunctions)
  - Dedicated cascade chain that implements high-speed, high-fan-in logic functions (automatically used by software tools and megafunctions)
  - Tri-state emulation that implements internal tri-state buses
  - Up to six global clock signals and four global clear signals

▩ Powerful I/O pins
  - Individual tri-state output enable control for each pin
  - Open-drain option on each I/O pin
  - Programmable output slew-rate control to reduce switching noise
  - Clamp to V_{CCIO} user-selectable on a pin-by-pin basis
  - Supports hot-socketing

- Software design support and automatic place-and-route provided by Altera's MAX+PLUS® II development system for Windows-based PCs and Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations, and the Quartus™ development system for Windows-based PCs and Sun SPARCstation and HP 9000 Series 700 workstations
- Flexible package options
  - Available in a variety of packages with 144 to 672 pins, including the innovative FineLine BGA™ packages (see Tables 3 and 4)
  - SameFrame™ pin compatibility with FLEX 10KA and FLEX 10KE devices across a range of device densities and pin counts
- Additional design entry and simulation support provided by EDIF 2 0 0 and 3 0 0 netlist files, library of parameterized modules (LPM), DesignWare components, Verilog HDL, VHDL, and other interfaces to popular EDA tools from manufacturers such as Cadence, Exemplar Logic, Mentor Graphics, OrCAD, Synopsys, Synplicity, VeriBest, and Viewlogic

Table 3. FLEX 10KE Package Options & I/O Pin Count *Notes (1), (2), (3)*

Device	144-Pin TQFP	208-Pin PQFP	240-Pin PQFP RQFP	256-Pin FineLine BGA	356-Pin BGA	484-Pin FineLine BGA	599-Pin PGA	600-Pin BGA	672-Pin FineLine BGA
EPF10K30E	102	147		176		220			220 *(4)*
EPF10K50E	102	147	189	191	220	254			254 *(4)*
EPF10K50S	102	147	189	191	220	254			254 *(4)*
EPF10K100B		147	189	191					
EPF10K100E		147	189	191	274	338			338 *(4)*
EPF10K130E			186		274	369		424	413
EPF10K200E							470	470	470
EPF10K200S			182		274	369	470	470	470

Notes:

(1) Contact Altera Customer Marketing for up-to-date information on package availability.

(2) FLEX 10KE device package types include thin quad flat pack (TQFP), plastic quad flat pack (PQFP), pin-grid array (PGA), and ball-grid array (BGA) packages.

(3) Devices in the same package are pin-compatible, although some devices have more I/O pins than others. When planning device migration, use the I/O pins that are common to all devices. The MAX+PLUS II software versions 9.1 and higher provide features to help designers use only the common pins.

(4) This option will be supported with a 484-pin FineLine BGA package. By using SameFrame pin migration, all FineLine BGA packages are pin-compatible. For example, a board can be designed to support 256-pin, 484-pin, and 672-pin FineLine BGA packages. The Quartus and MAX+PLUS II software automatically avoids conflicting pins when future migration is set.

Table 4. FLEX 10KE Package Sizes

Device	144-Pin TQFP	208-Pin PQFP	240-Pin PQFP RQFP	256-Pin FineLine βGA	356-Pin βGA	484-Pin FineLine βGA	599-Pin PGA	600-Pin βGA	672-Pin FineLine βGA
Pitch (mm)	0.50	0.50	0.50	1.0	1.27	1.0	–	1.27	1.0
Area (mm^2)	484	936	1,197	289	1,225	529	3,904	2,025	729
Length × width (mm × mm)	22 × 22	30.6 × 30.6	34.6 × 34.6	17 × 17	35 × 35	23 × 23	62.5 × 62.5	45 × 45	27 × 27

General Description

Altera FLEX 10KE devices are enhanced versions of the FLEX 10K device family. Based on reconfigurable CMOS SRAM elements, the Flexible Logic Element MatriX (FLEX) architecture incorporates all features necessary to implement common gate array megafunctions. With up to 200,000 gates, FLEX 10KE devices provide the density, speed, and features to integrate entire systems, including multiple 32-bit buses, into a single device.

The ability to reconfigure FLEX 10KE devices enables 100% testing prior to shipment and allows the designer to focus on simulation and design verification. FLEX 10KE reconfigurability eliminates inventory management for gate array designs and generation of test vectors for fault coverage.

Table 5 shows FLEX 10KE performance for some common designs. All performance values were obtained with Synopsys DesignWare or LPM functions. Special design techniques are not required to implement the applications; the designer simply infers or instantiates a function in a Verilog HDL, VHDL, Altera Hardware Description Language (AHDL), or schematic design file.

Table 5. FLEX 10KE Performance						
Application	**Resources Used**		**Performance**			**Units**
	LEs	**EABs**	**Speed Grade**			
			-1	**-2**	**-3**	
16-bit loadable counter	16	0	200	188	128	MHz
16-bit accumulator	16	0	200	188	128	MHz
16-to-1 multiplexer *(1)*	10	0	3.2	4.3	5.5	ns
16-bit multiplier with 3-stage pipeline *(2)*	544	0	93	86	64	MHz
256 × 16 RAM read cycle speed *(2)*	0	1	212	181	131	MHz
256 × 16 RAM write cycle speed *(2)*	0	1	142	128	94	MHz

Notes:

(1) This application uses combinatorial inputs and outputs.

(2) This application uses registered inputs and outputs.

Table 6 shows FLEX 10KE performance for more complex designs. These designs are available as Altera MegaCore™ functions.

Table 6. FLEX 10KE Performance for Complex Designs					
Application	**LEs Used**	**Performance**			**Units**
		Speed Grade			
		-1	**-2**	**-3**	
16-bit, 8-tap parallel finite impulse response (FIR) filter	420	185	175	122	MSPS
8-bit, 512-point fast Fourier transform (FFT) function	1,854	47.4	57.8	76.5	µs
		100	82	62	MHz
`a16450` universal asynchronous receiver/transmitter (UART)	342	66	57	44	MHz

Similar to the FLEX 10KE architecture, embedded gate arrays are the fastest-growing segment of the gate array market. As with standard gate arrays, embedded gate arrays implement general logic in a conventional "sea-of-gates" architecture. Additionally, embedded gate arrays have dedicated die areas for implementing large, specialized functions. By embedding functions in silicon, embedded gate arrays reduce die area and increase speed when compared to standard gate arrays. While embedded megafunctions typically cannot be customized, FLEX 10KE devices are programmable, providing the designer with full control over embedded megafunctions and general logic while facilitating iterative design changes during debugging.

Each FLEX 10KE device contains an embedded array and a logic array. The embedded array is used to implement a variety of memory functions or complex logic functions, such as digital signal processing (DSP), wide data-path manipulation, microcontroller applications, and data-transformation functions. The logic array performs the same function as the sea-of-gates in the gate array and is used to implement general logic, such as counters, adders, state machines, and multiplexers. The combination of embedded and logic arrays provides the high performance and high density of embedded gate arrays, enabling designers to implement an entire system on a single device.

FLEX 10KE devices are configured at system power-up with data stored in an Altera serial configuration device or provided by a system controller. Altera offers the EPC1, EPC2, and EPC1441 configuration devices, which configure FLEX 10KE devices via a serial data stream. Configuration data can also be downloaded from system RAM or via the Altera BitBlaster™, ByteBlaster™, or ByteBlasterMV™ download cables. (The ByteBlaster cable is obsolete and is replaced by the ByteBlasterMV cable, which can program and configure 2.5-V, 3.3-V, and 5.0-V devices.) After a FLEX 10KE device has been configured, it can be reconfigured in-circuit by resetting the device and loading new data. Because reconfiguration requires less than 276 ms, real-time changes can be made during system operation.

FLEX 10KE devices contain an interface that permits microprocessors to configure FLEX 10KE devices serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat a FLEX 10KE device as memory and configure it by writing to a virtual memory location, making it easy to reconfigure the device.

For more information, see the following documents:

- *Configuration Devices for APEX & FLEX Devices Data Sheet*
- *BitBlaster Serial Download Cable Data Sheet*
- *ByteBlasterMV Parallel Port Download Cable Data Sheet*

FLEX 10KE devices are supported by the Quartus and MAX+PLUS II development systems, which are integrated packages that offer schematic, text (including AHDL), and waveform design entry, compilation and logic synthesis, full simulation and worst-case timing analysis, and device configuration. The Quartus and MAX+PLUS II software provides EDIF 2 0 0 and 3 0 0, LPM, VHDL, Verilog HDL, and other interfaces for additional design entry and simulation support from other industry-standard PC- and UNIX workstation-based EDA tools.

The Quartus and MAX+PLUS II software works easily with common gate array EDA tools for synthesis and simulation. For example, the MAX+PLUS II software can generate Verilog HDL files for simulation with tools such as Cadence Verilog-XL. Additionally, the Quartus and MAX+PLUS II software contains EDA libraries that use device-specific features such as carry chains which are used for fast counter and arithmetic functions. For instance, the Synopsys Design Compiler library supplied with the Quartus and MAX+PLUS II development systems include DesignWare functions that are optimized for the FLEX 10KE architecture.

The MAX+PLUS II development system runs on Windows-based PCs and Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations, and the Quartus development system runs on Windows-based PCs, and Sun SPARCstation and HP 9000 Series 700 workstations.

See the *MAX+PLUS II Programmable Logic Development System & Software Data Sheet* and the *Quartus Programmable Logic Development System & Software Data Sheet* for more information.

Functional Description

Each FLEX 10KE device contains an enhanced embedded array to implement memory and specialized logic functions, and a logic array to implement general logic.

The embedded array consists of a series of EABs. When implementing memory functions, each EAB provides 4,096 bits, which can be used to create RAM, ROM, dual-port RAM, or first-in first-out (FIFO) functions. When implementing logic, each EAB can contribute 100 to 600 gates towards complex logic functions, such as multipliers, microcontrollers, state machines, and DSP functions. EABs can be used independently, or multiple EABs can be combined to implement larger functions.

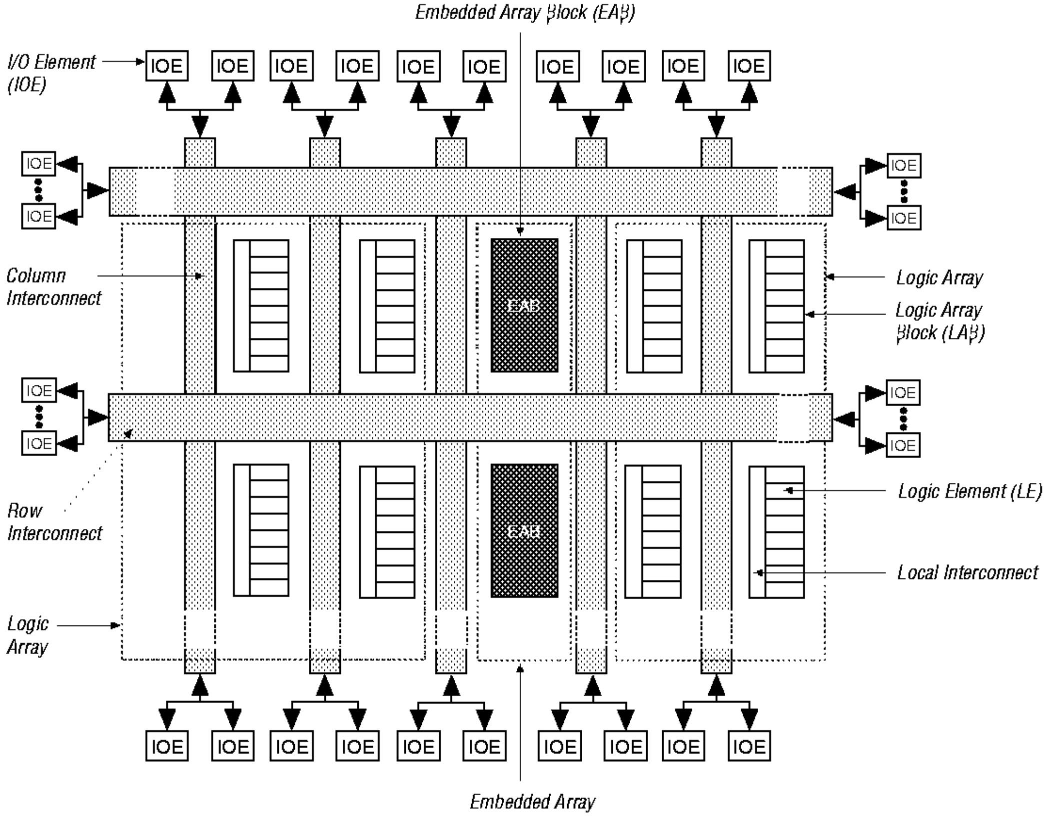
The logic array consists of logic array blocks (LABs). Each LAB contains eight LEs and a local interconnect. An LE consists of a 4-input look-up table (LUT), a programmable flipflop, and dedicated signal paths for carry and cascade functions. The eight LEs can be used to create medium-sized blocks of logic—such as 8-bit counters, address decoders, or state machines—or combined across LABs to create larger logic blocks. Each LAB represents about 96 usable gates of logic.

Signal interconnections within FLEX 10KE devices (as well as to and from device pins) are provided by the FastTrack Interconnect routing structure, which is a series of fast, continuous row and column channels that run the entire length and width of the device.

Each I/O pin is fed by an I/O element (IOE) located at the end of each row and column of the FastTrack Interconnect routing structure. Each IOE contains a bidirectional I/O buffer and a flipflop that can be used as either an output or input register to feed input, output, or bidirectional signals. When used with a dedicated clock pin, these registers provide exceptional performance. As inputs, they provide setup times as low as 2.5 ns and hold times of 0 ns. As outputs, these registers provide clock-to-output times as low as 3.7 ns. IOEs provide a variety of features, such as JTAG BST support, slew-rate control, tri-state buffers, and open-drain outputs.

Figure 1 shows a block diagram of the FLEX 10KE architecture. Each group of LEs is combined into an LAB; groups of LABs are arranged into rows and columns. Each row also contains a single EAB. The LABs and EABs are interconnected by the FastTrack Interconnect routing structure. IOEs are located at the end of each row and column of the FastTrack Interconnect routing structure.

Figure 1. FLEX 10KE Device Block Diagram



FLEX 10KE devices provide six dedicated inputs that drive the flipflops' control inputs and ensure the efficient distribution of high-speed, low-skew (less than 1.5 ns) control signals. These signals use dedicated routing channels that provide shorter delays and lower skews than the FastTrack Interconnect routing structure. Four of the dedicated inputs drive four global signals. These four global signals can also be driven by internal logic, providing an ideal solution for a clock divider or an internally generated asynchronous clear signal that clears many registers in the device.

Embedded Array Block

The EAB is a flexible block of RAM with registers on the input and output ports that is used to implement common gate array megafunctions. Because it is large and flexible, the EAB is suitable for functions such as multipliers, vector scalars, and error correction circuits. These functions can be combined in applications such as digital filters and microcontrollers.

Logic functions are implemented by programming the EAB with a read-only pattern during configuration, thereby creating a large LUT. With LUTs, combinatorial functions are implemented by looking up the results, rather than by computing them. This implementation of combinatorial functions can be faster than using algorithms implemented in general logic, a performance advantage that is further enhanced by the fast access times of EABs. The large capacity of EABs enables designers to implement complex functions in one logic level without the routing delays associated with linked LEs or field-programmable gate array (FPGA) RAM blocks. For example, a single EAB can implement any function with 8 inputs and 16 outputs. Parameterized functions such as LPM functions can take advantage of the EAB automatically.

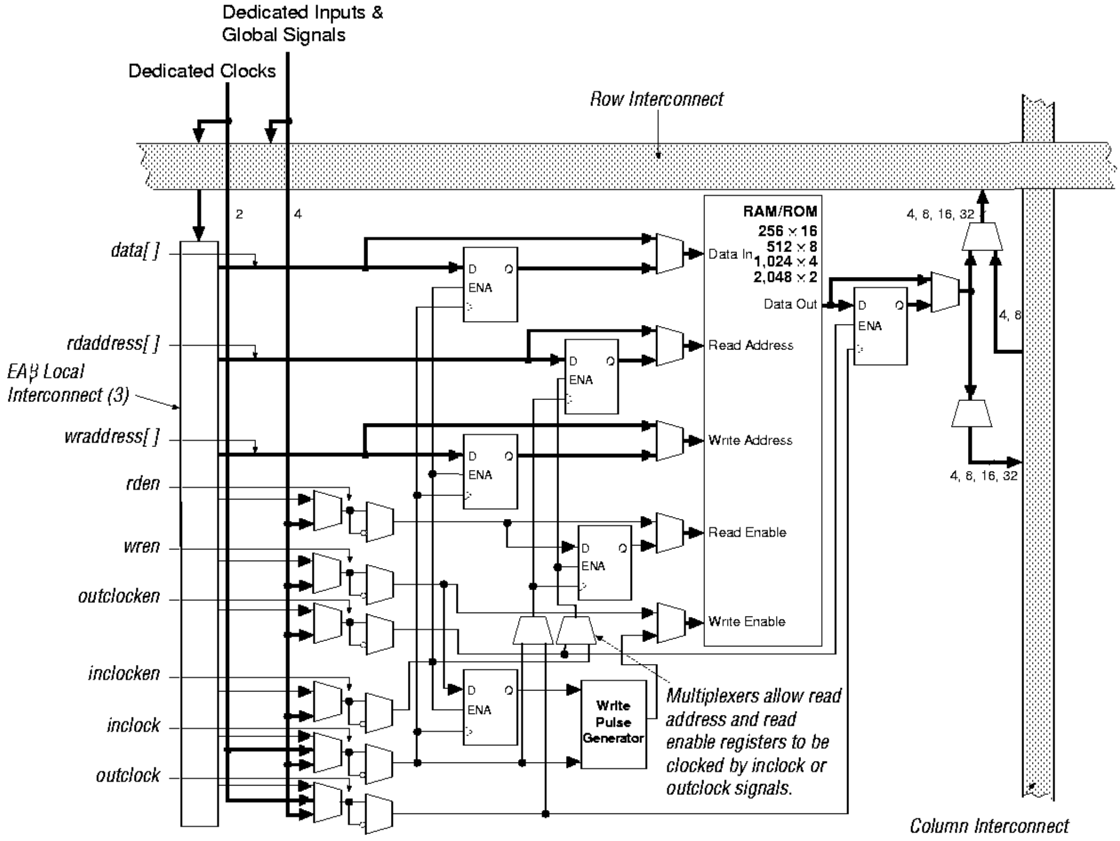
The FLEX 10KE EAB provides advantages over FPGAs, which implement on-board RAM as arrays of small, distributed RAM blocks. These small FPGA RAM blocks must be connected together to make RAM blocks of manageable size. The RAM blocks are connected together using multiplexers implemented with more logic blocks. These extra multiplexers cause extra delay, which slows down the RAM block. FPGA RAM blocks are also prone to routing problems because small blocks of RAM must be connected together to make larger blocks. In contrast, EABs can be used to implement large, dedicated blocks of RAM that eliminate these timing and routing concerns.

The FLEX 10KE enhanced EAB adds dual-port capability to the existing EAB structure. The dual-port structure is ideal for FIFO buffers with one or two clocks. The FLEX 10KE EAB can also support up to 16-bit-wide RAM blocks and is backward-compatible with any design containing FLEX 10K EABs. The FLEX 10KE EAB can act in dual-port or single-port mode. When in dual-port mode, separate clocks may be used for EAB read and write sections, which allows the EAB to be written and read at different rates. It also has separate synchronous clock enable signals for the EAB read and write sections, which allow independent control of these sections.

The EAB can also be used for bidirectional, dual-port memory applications where two ports read or write simultaneously. To implement this type of dual-port memory, two EABs are used to support two simultaneous read or writes.

Alternatively, one clock and clock enable can be used to control the input registers of the EAB, while a different clock and clock enable control the output registers (see Figure 2).

Figure 2. FLEX 10KE Device in Dual-Port RAM Mode *Notes (1), (2)*



Notes:

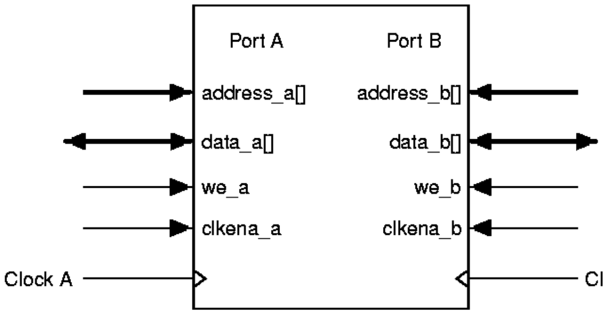
(1) All registers can be asynchronously cleared by EAB local interconnect signals, global signals, or the chip-wide reset.

(2) The EPF10K100B device does not offer dual-port RAM mode.

(3) EPF10K30E and EPF10K50E devices have 88 EAB local interconnect channels; EPF10K100E, EPF10K130E, and EPF10K200E devices have 104 EAB local interconnect channels.

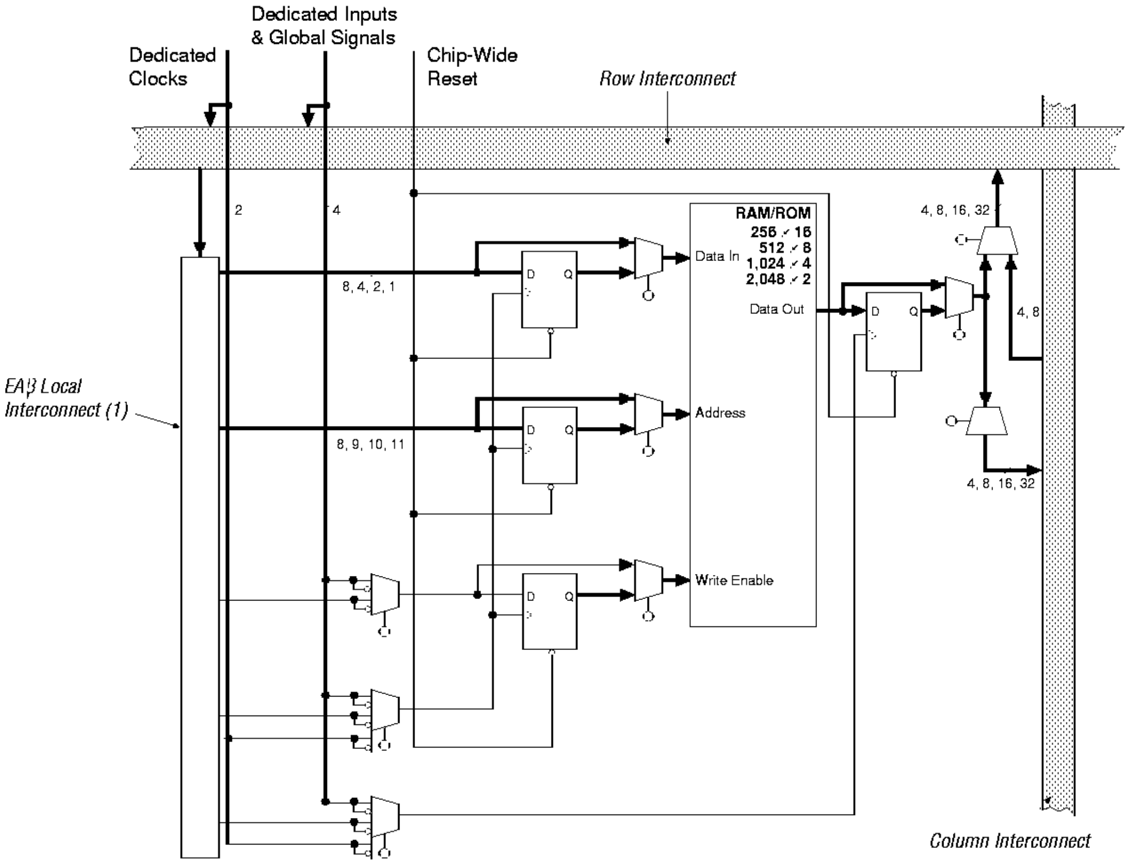
The EAB can also use Altera megafunctions to implement dual-port RAM applications where both ports can read or write, as shown in Figure 3.

Figure 3. FLEX 10KE EAB in Dual-Port RAM Mode



The FLEX 10KE EAB can be used in a single-port mode, which is useful for backward-compatibility with FLEX 10K designs (see Figure 4).

Figure 4. FLEX 10KE Device in Single-Port RAM Mode



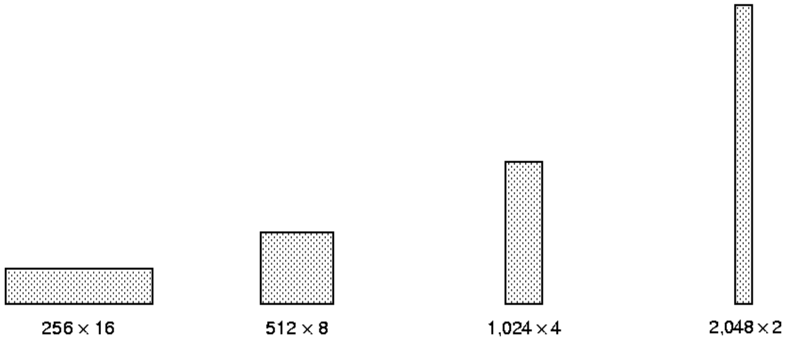
Note:

(1) EPF10K30E, EPF10K50E, and EPF10K50S devices have 88 EAB local interconnect channels; EPF10K100E, EPF10K100B, EPF10K130E, EPF10K200E, and EPF10K200S devices have 104 EAB local interconnect channels.

EABs can be used to implement synchronous RAM, which is easier to use than asynchronous RAM. A circuit using asynchronous RAM must generate the RAM write enable signal, while ensuring that its data and address signals meet setup and hold time specifications relative to the write enable signal. In contrast, the EAB's synchronous RAM generates its own write enable signal and is self-timed with respect to the input or write clock. A circuit using the EAB's self-timed RAM must only meet the setup and hold time specifications of the global clock.

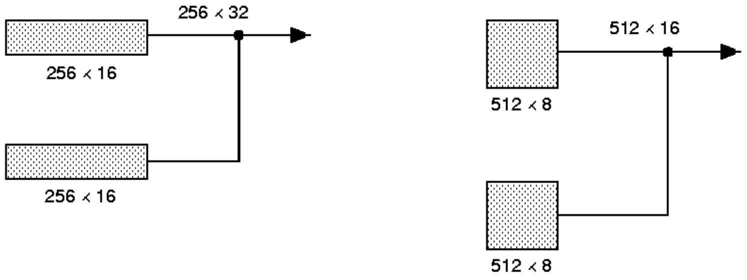
When used as RAM, each EAB can be configured in any of the following sizes: 256 × 16, 512 × 8, 1,024 × 4, or 2,048 × 2 (see Figure 5).

Figure 5. FLEX 10KE EAβ Memory Configurations



Larger blocks of RAM are created by combining multiple EABs. For example, two 256 × 16 RAM blocks can be combined to form a 256 × 32 block; two 512 × 8 RAM blocks can be combined to form a 512 × 16 block (see Figure 6).

Figure 6. Examples of Combining FLEX 10KE EAβs



If necessary, all EABs in a device can be cascaded to form a single RAM block. EABs can be cascaded to form RAM blocks of up to 2,048 words without impacting timing. The Quartus and MAX+PLUS II software automatically combines EABs to meet a designer's RAM specifications.

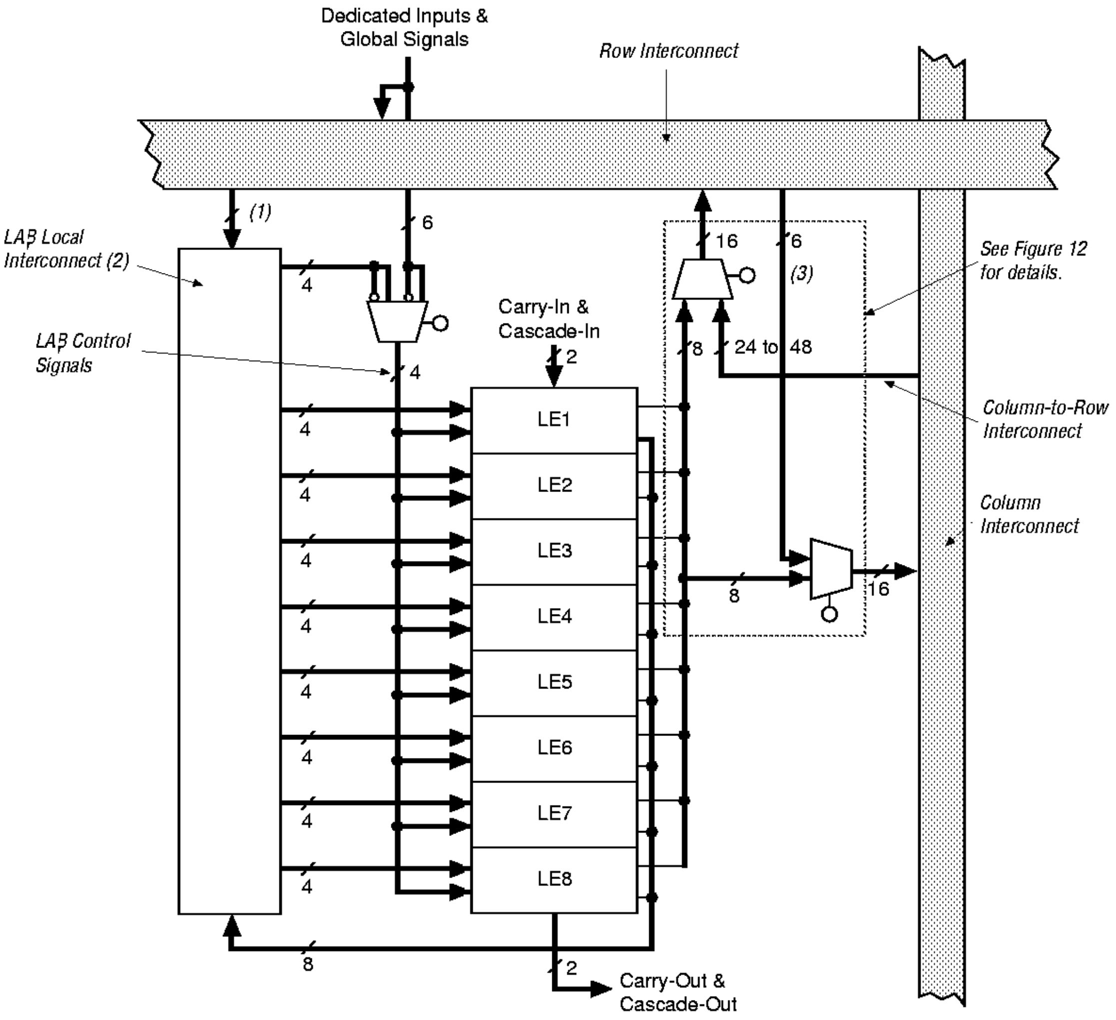
EABs provide flexible options for driving and controlling clock signals. Different clocks and clock enables can be used for reading and writing to the EAB. Registers can be independently inserted on the data input, EAB output, write address, write enable signals, read address, and read enable signals. The global signals and the EAB local interconnect can drive write enable, read enable, and clock enable signals. The global signals, dedicated clock pins, and EAB local interconnect can drive the EAB clock signals. Because the LEs drive the EAB local interconnect, the LEs can control write enable, read enable, clear, clock, and clock enable signals.

An EAB is fed by a row interconnect and can drive out to row and column interconnects. Each EAB output can drive up to two row channels and up to two column channels; the unused row channel can be driven by other LEs. This feature increases the routing resources available for EAB outputs (see Figures 2 and 4). The column interconnect, which is adjacent to the EAB, has twice as many channels as other columns in the device.

Logic Array Block

An LAB consists of eight LEs, their associated carry and cascade chains, LAB control signals, and the LAB local interconnect. The LAB provides the coarse-grained structure to the FLEX 10KE architecture, facilitating efficient routing with optimum device utilization and high performance (see Figure 7).

Figure 7. FLEX 10KE LAB



Notes:

(1) EPF10K30E, EPF10K50E, and EPF10K50S devices have 22 inputs to the LAB local interconnect channel from the row; EPF10K100E, EPF10K100B, EPF10K130E, EPF10K200E, and EPF10K200S devices have 26.

(2) EPF10K30E, EPF10K50E, and EPF10K50S devices have 30 LAB local interconnect channels; EPF10K100E, EPF10K100B, EPF10K130E, EPF10K200E, and EPF10K200S devices have 34.

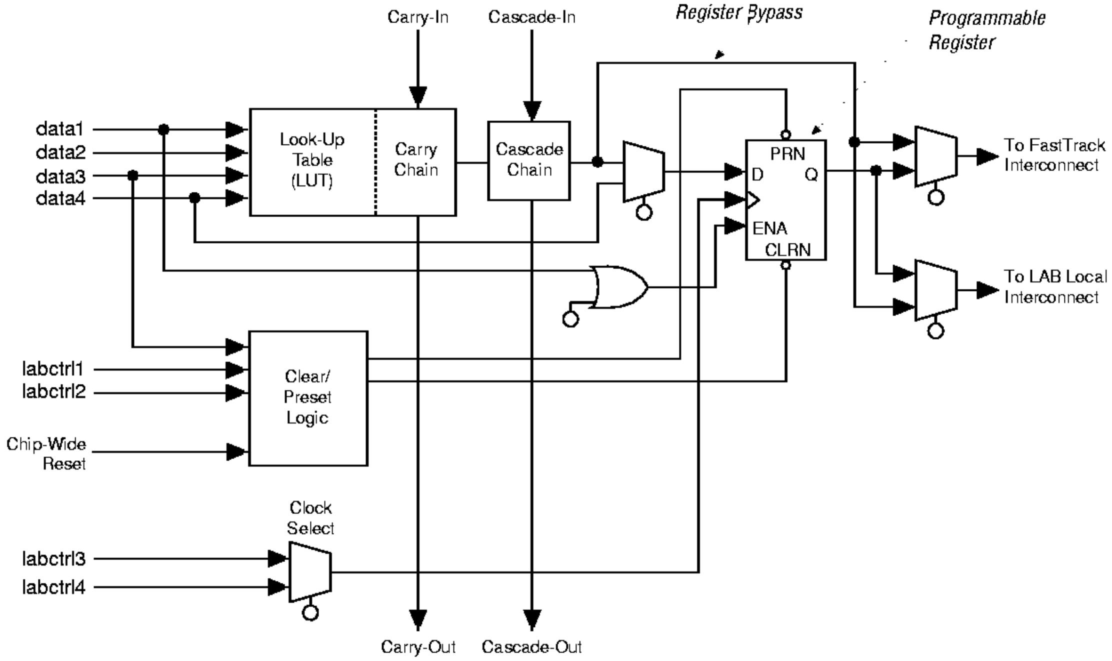
(3) In EPF10K100B devices, four row channels can drive column channels at each intersection.

Each LAB provides four control signals with programmable inversion that can be used in all eight LEs. Two of these signals can be used as clocks, the other two can be used for clear/preset control. The LAB clocks can be driven by the dedicated clock input pins, global signals, I/O signals, or internal signals via the LAB local interconnect. The LAB preset and clear control signals can be driven by the global signals, I/O signals, or internal signals via the LAB local interconnect. The global control signals are typically used for global clock, clear, or preset signals because they provide asynchronous control with very low skew across the device. If logic is required on a control signal, it can be generated in one or more LE in any LAB and driven into the local interconnect of the target LAB. In addition, the global control signals can be generated from LE outputs.

Logic Element

The LE, the smallest unit of logic in the FLEX 10KE architecture, has a compact size that provides efficient logic utilization. Each LE contains a 4-input LUT, which is a function generator that can quickly compute any function of four variables. In addition, each LE contains a programmable flipflop with a synchronous clock enable, a carry chain, and a cascade chain. Each LE drives both the local and the FastTrack Interconnect routing structure (see Figure 8).

Figure 8. FLEX 10KE Logic Element



The programmable flipflop in the LE can be configured for D, T, JK, or SR operation. The clock, clear, and preset control signals on the flipflop can be driven by global signals, general-purpose I/O pins, or any internal logic. For combinatorial functions, the flipflop is bypassed and the output of the LUT drives the output of the LE.

The LE has two outputs that drive the interconnect: one drives the local interconnect and the other drives either the row or column FastTrack Interconnect routing structure. The two outputs can be controlled independently. For example, the LUT can drive one output while the register drives the other output. This feature, called register packing, can improve LE utilization because the register and the LUT can be used for unrelated functions.

The FLEX 10KE architecture provides two types of dedicated high-speed data paths that connect adjacent LEs without using local interconnect paths: carry chains and cascade chains. The carry chain supports high-speed counters and adders and the cascade chain implements wide-input functions with minimum delay. Carry and cascade chains connect all LEs in a LAB as well as all LABs in the same row. Intensive use of carry and cascade chains can reduce routing flexibility. Therefore, the use of these chains should be limited to speed-critical portions of a design.

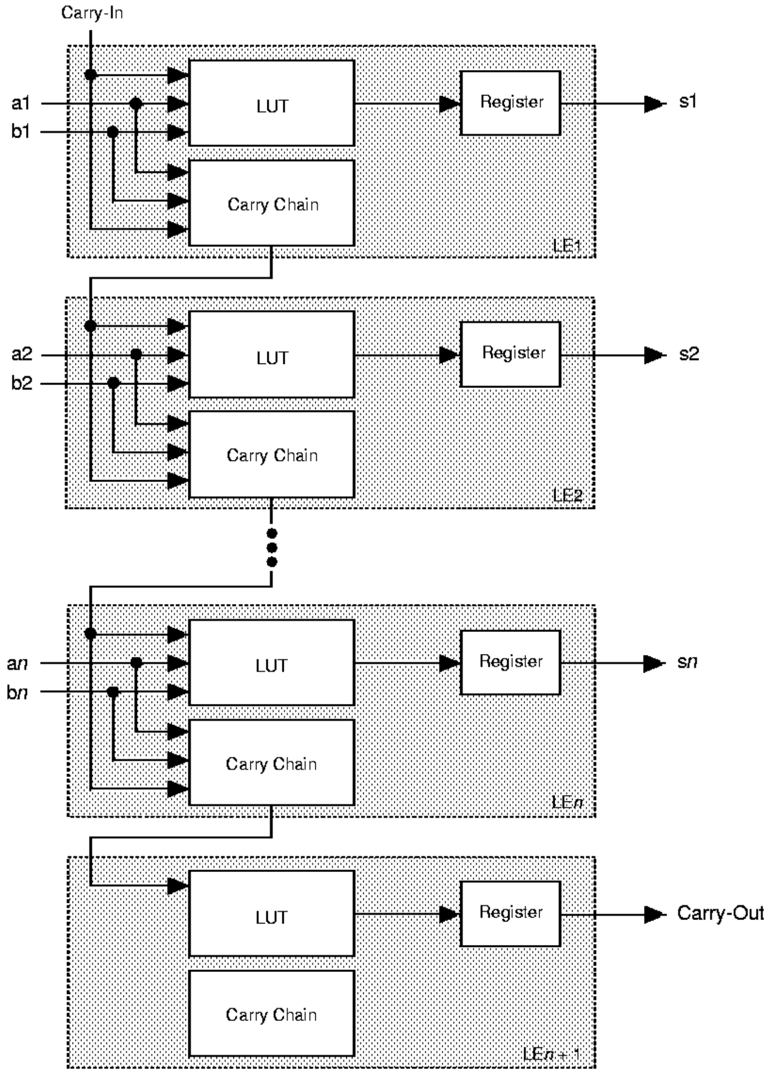
Carry Chain

The carry chain provides a very fast (as low as 0.2 ns) carry-forward function between LEs. The carry-in signal from a lower-order bit drives forward into the higher-order bit via the carry chain, and feeds into both the LUT and the next portion of the carry chain. This feature allows the FLEX 10KE architecture to implement high-speed counters, adders, and comparators of arbitrary width efficiently. Carry chain logic can be created automatically by the Quartus and MAX+PLUS II Compilers during design processing, or manually by the designer during design entry. Parameterized functions such as LPM and DesignWare functions automatically take advantage of carry chains.

Carry chains longer than eight LEs are automatically implemented by linking LABs together. For enhanced fitting, a long carry chain skips alternate LABs in a row. A carry chain longer than one LAB skips either from even-numbered LAB to even-numbered LAB, or from odd-numbered LAB to odd-numbered LAB. For example, the last LE of the first LAB in a row carries to the first LE of the third LAB in the row. The carry chain does not cross the EAB at the middle of the row. For instance, in the EPF10K50E device, the carry chain stops at the eighteenth LAB and a new one begins at the nineteenth LAB.

Figure 9 shows how an *n*-bit full adder can be implemented in *n* + 1 LEs with the carry chain. One portion of the LUT generates the sum of two bits using the input signals and the carry-in signal; the sum is routed to the output of the LE. The register can be bypassed for simple adders or used for an accumulator function. Another portion of the LUT and the carry chain logic generates the carry-out signal, which is routed directly to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it can be used as a general-purpose signal.

Figure 9. FLEX 10KE Carry Chain Operation (n-βit Full Adder)



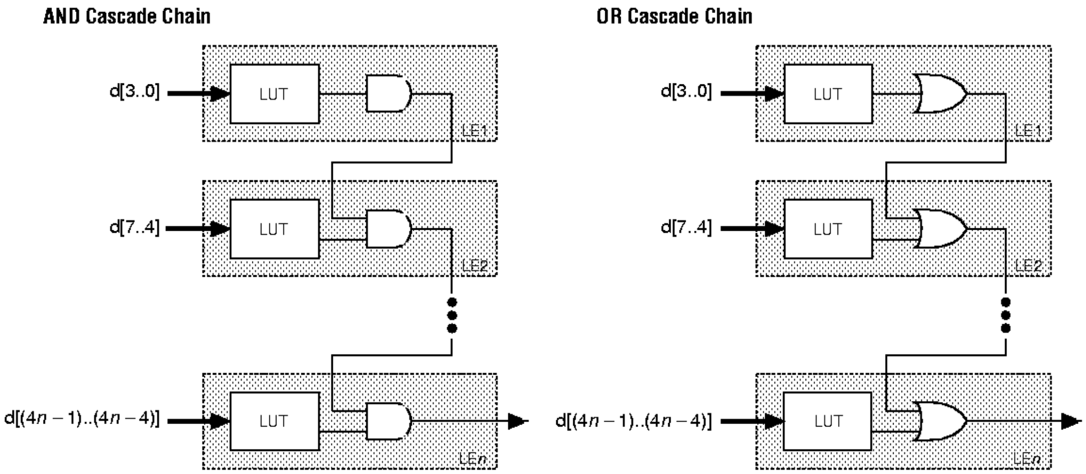
Cascade Chain

With the cascade chain, the FLEX 10KE architecture can implement functions that have a very wide fan-in. Adjacent LUTs can be used to compute portions of the function in parallel; the cascade chain serially connects the intermediate values. The cascade chain can use a logical AND or logical OR (via De Morgan's inversion) to connect the outputs of adjacent LEs. With a delay as low as 0.6 ns per LE, each additional LE provides four more inputs to the effective width of a function. Cascade chain logic can be created automatically by the MAX+PLUS II Compiler during design processing, or manually by the designer during design entry.

Cascade chains longer than eight bits are implemented automatically by linking several LABs together. For easier routing, a long cascade chain skips every other LAB in a row. A cascade chain longer than one LAB skips either from even-numbered LAB to even-numbered LAB, or from odd-numbered LAB to odd-numbered LAB (e.g., the last LE of the first LAB in a row cascades to the first LE of the third LAB). The cascade chain does not cross the center of the row (e.g., in the EPF10K50E device, the cascade chain stops at the eighteenth LAB and a new one begins at the nineteenth LAB). This break is due to the EAB's placement in the middle of the row.

Figure 10 shows how the cascade function can connect adjacent LEs to form functions with a wide fan-in. These examples show functions of 4*n* variables implemented with *n* LEs. The LE delay is 1.3 ns; the cascade chain delay is 0.6 ns. With the cascade chain, 3.1 ns are needed to decode a 16-bit address.

Figure 10. FLEX 10KE Cascade Chain Operation



AND Cascade Chain

OR Cascade Chain

LE Operating Modes

The FLEX 10KE LE can operate in the following four modes:

- Normal mode
- Arithmetic mode
- Up/down counter mode
- Clearable counter mode

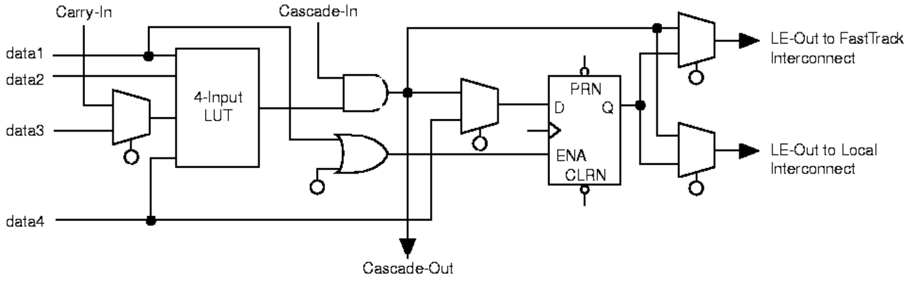
Each of these modes uses LE resources differently. In each mode, seven available inputs to the LE—the four data inputs from the LAB local interconnect, the feedback from the programmable register, and the carry-in and cascade-in from the previous LE—are directed to different destinations to implement the desired logic function. Three inputs to the LE provide clock, clear, and preset control for the register. The Quartus and MAX+PLUS II software, in conjunction with parameterized functions such as LPM and DesignWare functions, automatically chooses the appropriate mode for common functions such as counters, adders, and multipliers. If required, the designer can also create special-purpose functions that use a specific LE operating mode for optimal performance.

The architecture provides a synchronous clock enable to the register in all four modes. The Quartus and MAX+PLUS II software can set `DATA1` to enable the register synchronously, providing easy implementation of fully synchronous designs.

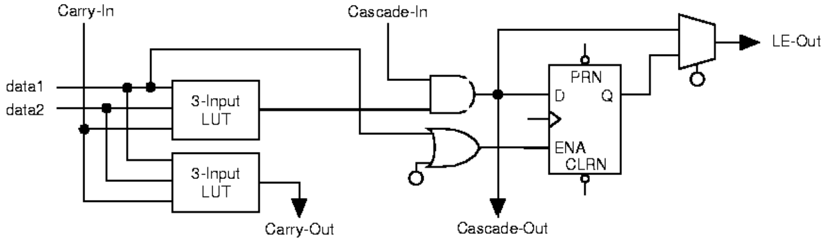
Figure 11 shows the LE operating modes.

Figure 11. FLEX 10KE LE Operating Modes

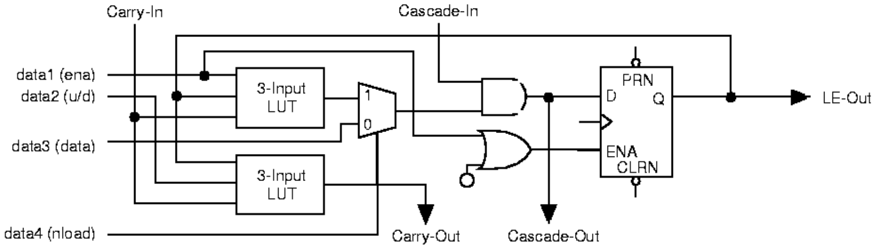
Normal Mode



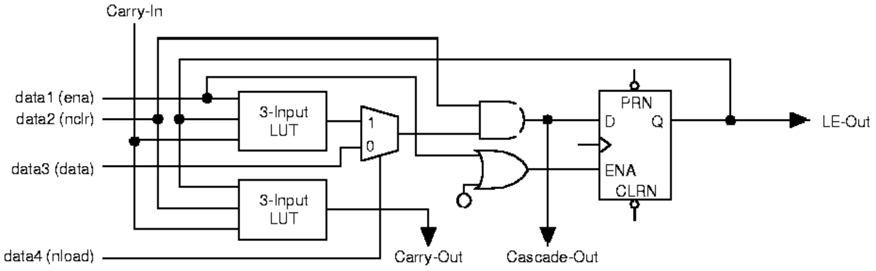
Arithmetic Mode



Up/Down Counter Mode



Clearable Counter Mode



Normal Mode

The normal mode is suitable for general logic applications and wide decoding functions that can take advantage of a cascade chain. In normal mode, four data inputs from the LAB local interconnect and the carry-in are inputs to a 4-input LUT. The MAX+PLUS II Compiler automatically selects the carry-in or the `DATA3` signal as one of the inputs to the LUT. The LUT output can be combined with the cascade-in signal to form a cascade chain through the cascade-out signal. Either the register or the LUT can be used to drive both the local interconnect and the FastTrack Interconnect routing structure at the same time.

The LUT and the register in the LE can be used independently (register packing). To support register packing, the LE has two outputs; one drives the local interconnect, and the other drives the FastTrack Interconnect routing structure. The `DATA4` signal can drive the register directly, allowing the LUT to compute a function that is independent of the registered signal; a 3-input function can be computed in the LUT, and a fourth independent signal can be registered. Alternatively, a 4-input function can be generated, and one of the inputs to this function can be used to drive the register. The register in a packed LE can still use the clock enable, clear, and preset signals in the LE. In a packed LE, the register can drive the FastTrack Interconnect routing structure while the LUT drives the local interconnect, or vice versa.

Arithmetic Mode

The arithmetic mode offers two 3-input LUTs that are ideal for implementing adders, accumulators, and comparators. One LUT computes a 3-input function; the other generates a carry output. As shown in Figure 11 on page 22, the first LUT uses the carry-in signal and two data inputs from the LAB local interconnect to generate a combinatorial or registered output. For example, in an adder, this output is the sum of three signals: `a`, `b`, and carry-in. The second LUT uses the same three signals to generate a carry-out signal, thereby creating a carry chain. The arithmetic mode also supports simultaneous use of the cascade chain.

Up/Down Counter Mode

The up/down counter mode offers counter enable, clock enable, synchronous up/down control, and data loading options. These control signals are generated by the data inputs from the LAB local interconnect, the carry-in signal, and output feedback from the programmable register. Two 3-input LUTs are used: one generates the counter data, and the other generates the fast carry bit. A 2-to-1 multiplexer provides synchronous loading. Data can also be loaded asynchronously with the clear and preset register control signals, without using the LUT resources.

Clearable Counter Mode

The clearable counter mode is similar to the up/down counter mode, but supports a synchronous clear instead of the up/down control. The clear function is substituted for the cascade-in signal in the up/down counter mode. Two 3-input LUTs are used: one generates the counter data, and the other generates the fast carry bit. Synchronous loading is provided by a 2-to-1 multiplexer. The output of this multiplexer is `AND`ed with a synchronous clear signal.

Internal Tri-State Emulation

Internal tri-state emulation provides internal tri-states without the limitations of a physical tri-state bus. In a physical tri-state bus, the tri-state buffers' output enable (`OE`) signals select which signal drives the bus. However, if multiple `OE` signals are active, contending signals can be driven onto the bus. Conversely, if no `OE` signals are active, the bus will float. Internal tri-state emulation resolves contending tri-state buffers to a low value and floating buses to a high value, thereby eliminating these problems. The MAX+PLUS II software automatically implements tri-state bus functionality with a multiplexer.

Clear & Preset Logic Control

Logic for the programmable register's clear and preset functions is controlled by the `DATA3`, `LABCTRL1`, and `LABCTRL2` inputs to the LE. The clear and preset control structure of the LE asynchronously loads signals into a register. Either `LABCTRL1` or `LABCTRL2` can control the asynchronous clear. Alternatively, the register can be set up so that `LABCTRL1` implements an asynchronous load. The data to be loaded is driven to `DATA3`; when `LABCTRL1` is asserted, `DATA3` is loaded into the register.

During compilation, the Quartus and MAX+PLUS II Compilers automatically select the best control signal implementation. Because the clear and preset functions are active-low, the Compiler automatically assigns a logic high to an unused clear or preset.

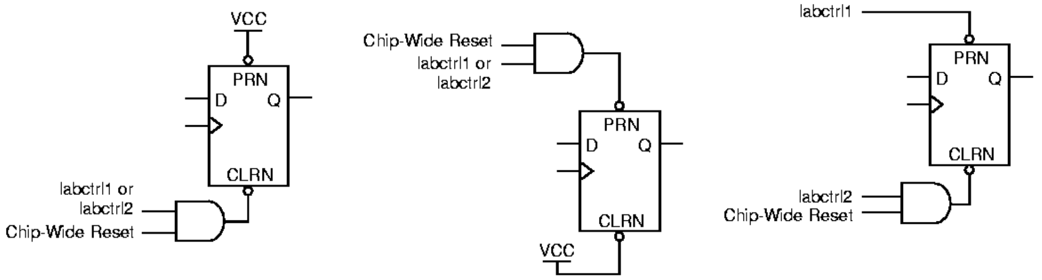
The clear and preset logic is implemented in one of the following six modes chosen during design entry:

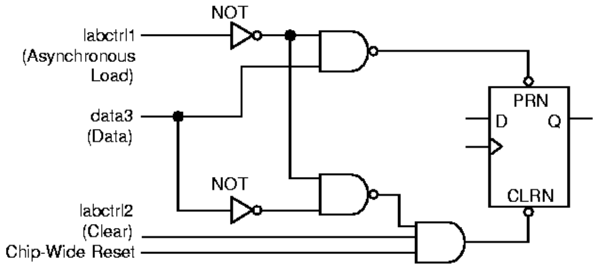
- Asynchronous clear
- Asynchronous preset
- Asynchronous clear and preset
- Asynchronous load with clear
- Asynchronous load with preset
- Asynchronous load without clear or preset

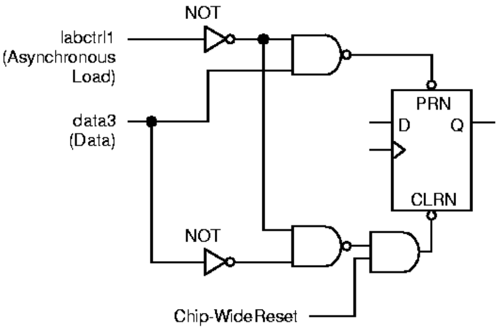
In addition to the six clear and preset modes, FLEX 10KE devices provide a chip-wide reset pin that can reset all registers in the device. Use of this feature is set during design entry. In any of the clear and preset modes, the chip-wide reset overrides all other signals. Registers with asynchronous presets may be preset when the chip-wide reset is asserted. Inversion can be used to implement the asynchronous preset. Figure 12 shows examples of how to setup the preset and clear inputs for the desired functionality.

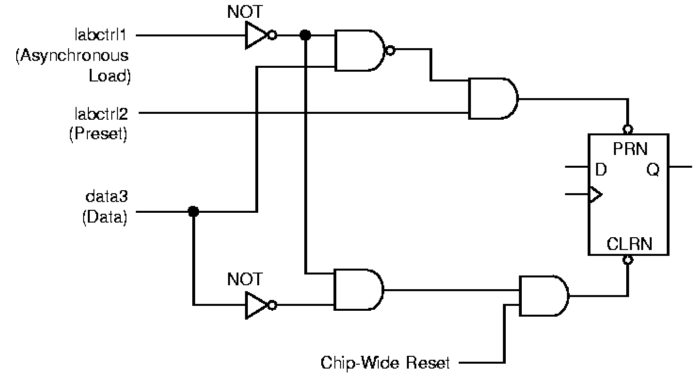
Figure 12. FLEX 10KE LE Clear & Preset Modes

Asynchronous Clear **Asynchronous Preset** **Asynchronous Preset & Clear**




Asynchronous Load with Clear


Asynchronous Load without Clear or Preset


Asynchronous Load with Preset

Asynchronous Clear

The flipflop can be cleared by either `LABCTRL1` or `LABCTRL2`. In this mode, the preset signal is tied to `VCC` to deactivate it.

Asynchronous Preset

An asynchronous preset is implemented as an asynchronous load, or with an asynchronous clear. If `DATA3` is tied to `VCC`, asserting `LABCTRL1` asynchronously loads a one into the register. Alternatively, the MAX+PLUS II software can provide preset control by using the clear and inverting the input and output of the register. Inversion control is available for the inputs to both LEs and IOEs. Therefore, if a register is preset by only one of the two `LABCTRL` signals, the `DATA3` input is not needed and can be used for one of the LE operating modes.

Asynchronous Preset & Clear

When implementing asynchronous clear and preset, `LABCTRL1` controls the preset and `LABCTRL2` controls the clear. `DATA3` is tied to `VCC`, so that asserting `LABCTRL1` asynchronously loads a one into the register, effectively presetting the register. Asserting `LABCTRL2` clears the register.

Asynchronous Load with Clear

When implementing an asynchronous load in conjunction with the clear, `LABCTRL1` implements the asynchronous load of `DATA3` by controlling the register preset and clear. `LABCTRL2` implements the clear by controlling the register clear; `LABCTRL2` does not have to feed the preset circuits.

Asynchronous Load with Preset

When implementing an asynchronous load in conjunction with preset, the MAX+PLUS II software provides preset control by using the clear and inverting the input and output of the register. Asserting `LABCTRL2` presets the register, while asserting `LABCTRL1` loads the register. The MAX+PLUS II software inverts the signal that drives `DATA3` to account for the inversion of the register's output.

Asynchronous Load without Preset or Clear

When implementing an asynchronous load without preset or clear, `LABCTRL1` implements the asynchronous load of `DATA3` by controlling the register preset and clear.

FastTrack Interconnect Routing Structure

In the FLEX 10KE architecture, connections between LEs, EABs, and device I/O pins are provided by the FastTrack Interconnect routing structure, which is a series of continuous horizontal and vertical routing channels that traverses the device. This global routing structure provides predictable performance, even in complex designs. In contrast, the segmented routing in FPGAs requires switch matrices to connect a variable number of routing paths, increasing the delays between logic resources and reducing performance.

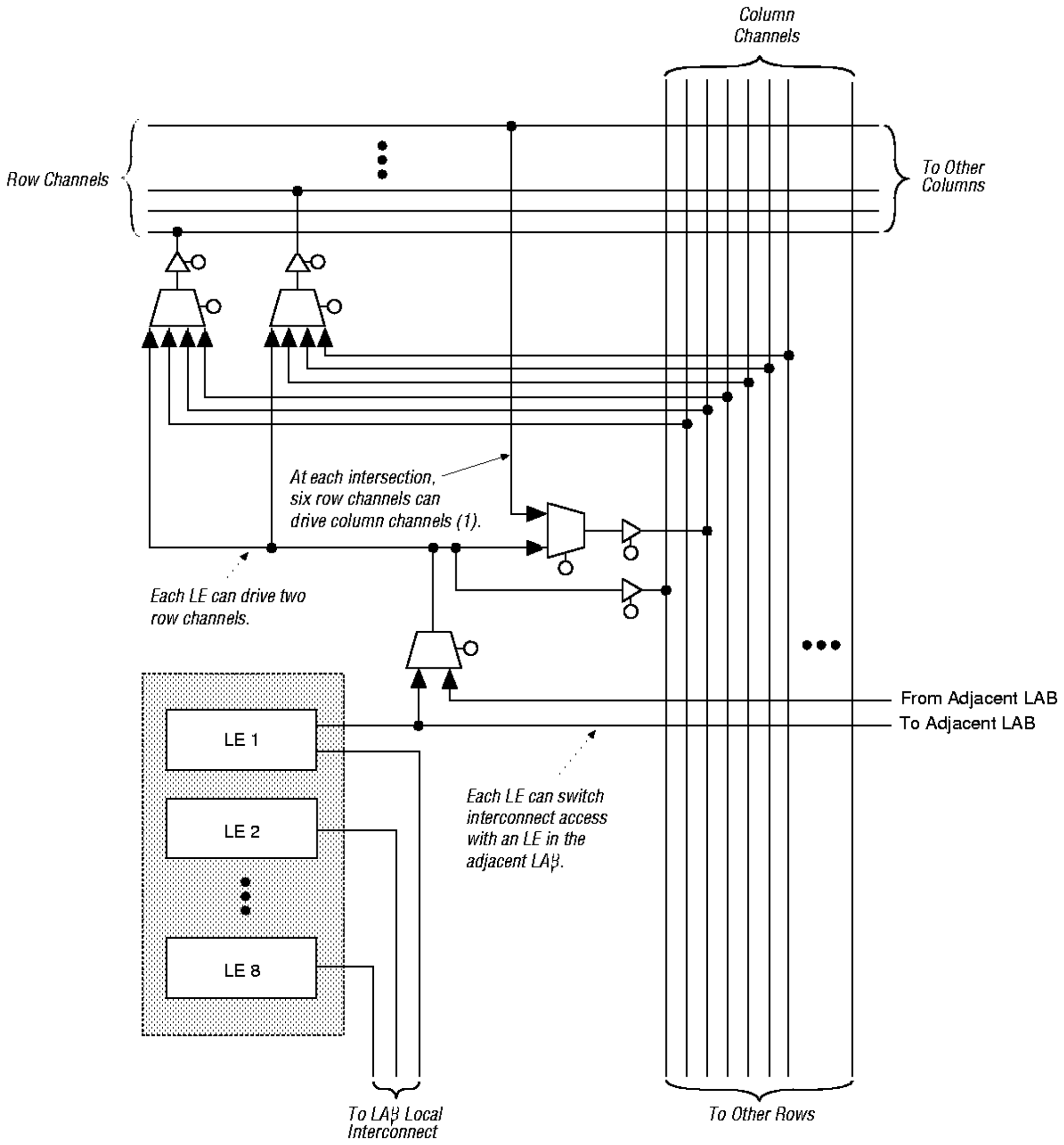
The FastTrack Interconnect routing structure consists of row and column interconnect channels that span the entire device. Each row of LABs is served by a dedicated row interconnect. The row interconnect can drive I/O pins and feed other LABs in the row. The column interconnect routes signals between rows and can drive I/O pins.

Row channels drive into the LAB or EAB local interconnect. The row signal is buffered at every LAB or EAB to reduce the effect of fan-out on delay. A row channel can be driven by an LE or by one of three column channels. These four signals feed dual 4-to-1 multiplexers that connect to two specific row channels. These multiplexers, which are connected to each LE, allow column channels to drive row channels even when all eight LEs in a LAB drive the row interconnect.

Each column of LABs or EABs is served by a dedicated column interconnect. The column interconnect that serves the EABs has twice as many channels as other column interconnects. The column interconnect can then drive I/O pins or another row's interconnect to route the signals to other LABs or EABs in the device. A signal from the column interconnect, which can be either the output of a LE or an input from an I/O pin, must be routed to the row interconnect before it can enter a LAB or EAB. Each row channel that is driven by an IOE or EAB can drive one specific column channel.

Access to row and column channels can be switched between LEs in adjacent pairs of LABs. For example, a LE in one LAB can drive the row and column channels normally driven by a particular LE in the adjacent LAB in the same row, and vice versa. This flexibility enables routing resources to be used more efficiently (see Figure 13).

Figure 13. FLEX 10KE LAB Connections to Row & Column Interconnect



Note:

(1) In EPF10K100B devices, four row channels can drive column channels at each intersection.

For improved routing, the row interconnect consists of a combination of full-length and half-length channels. The full-length channels connect to all LABs in a row; the half-length channels connect to the LABs in half of the row. The EAB can be driven by the half-length channels in the left half of the row and by the full-length channels. The EAB drives out to the full-length channels. In addition to providing a predictable, row-wide interconnect, this architecture provides increased routing resources. Two neighboring LABs can be connected using a half-row channel, thereby saving the other half of the channel for the other half of the row.

Table 7 summarizes the FastTrack Interconnect routing structure resources available in each FLEX 10KE device.

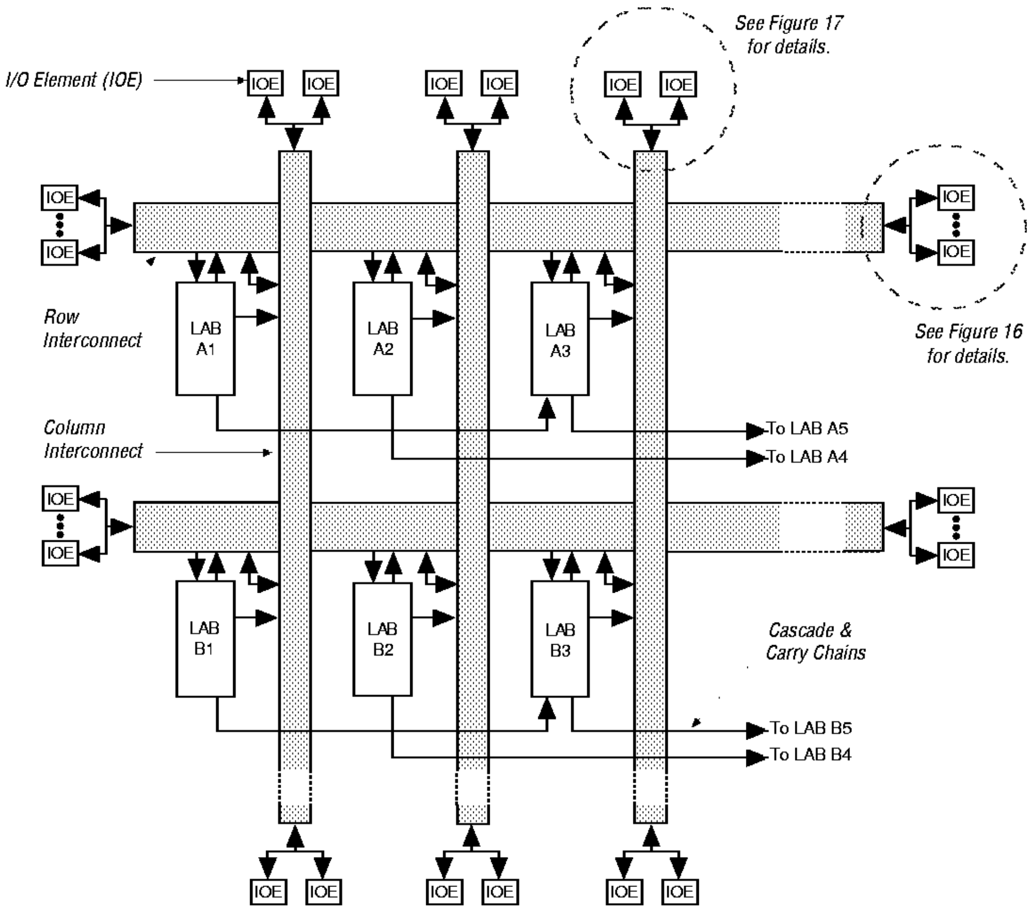
Table 7. FLEX 10KE FastTrack Interconnect Resources

Device	Rows	Channels per Row	Columns	Channels per Column
EPF10K30E	6	216	36	24
EPF10K50E EPF10K50S	10	216	36	24
EPF10K100B EPF10K100E	12	312	52	24
EPF10K130E	16	312	52	32
EPF10K200E EPF10K200S	24	312	52	48

In addition to general-purpose I/O pins, FLEX 10KE devices have six dedicated input pins that provide low-skew signal distribution across the device. These six inputs can be used for global clock, clear, preset, and peripheral output enable and clock enable control signals. These signals are available as control signals for all LABs and IOEs in the device. The dedicated inputs can also be used as general-purpose data inputs because they can feed the local interconnect of each LAB in the device.

Figure 14 shows the interconnection of adjacent LABs and EABs, with row, column, and local interconnects, as well as the associated cascade and carry chains. Each LAB is labeled according to its location: a letter represents the row and a number represents the column. For example, LAB B3 is in row B, column 3.

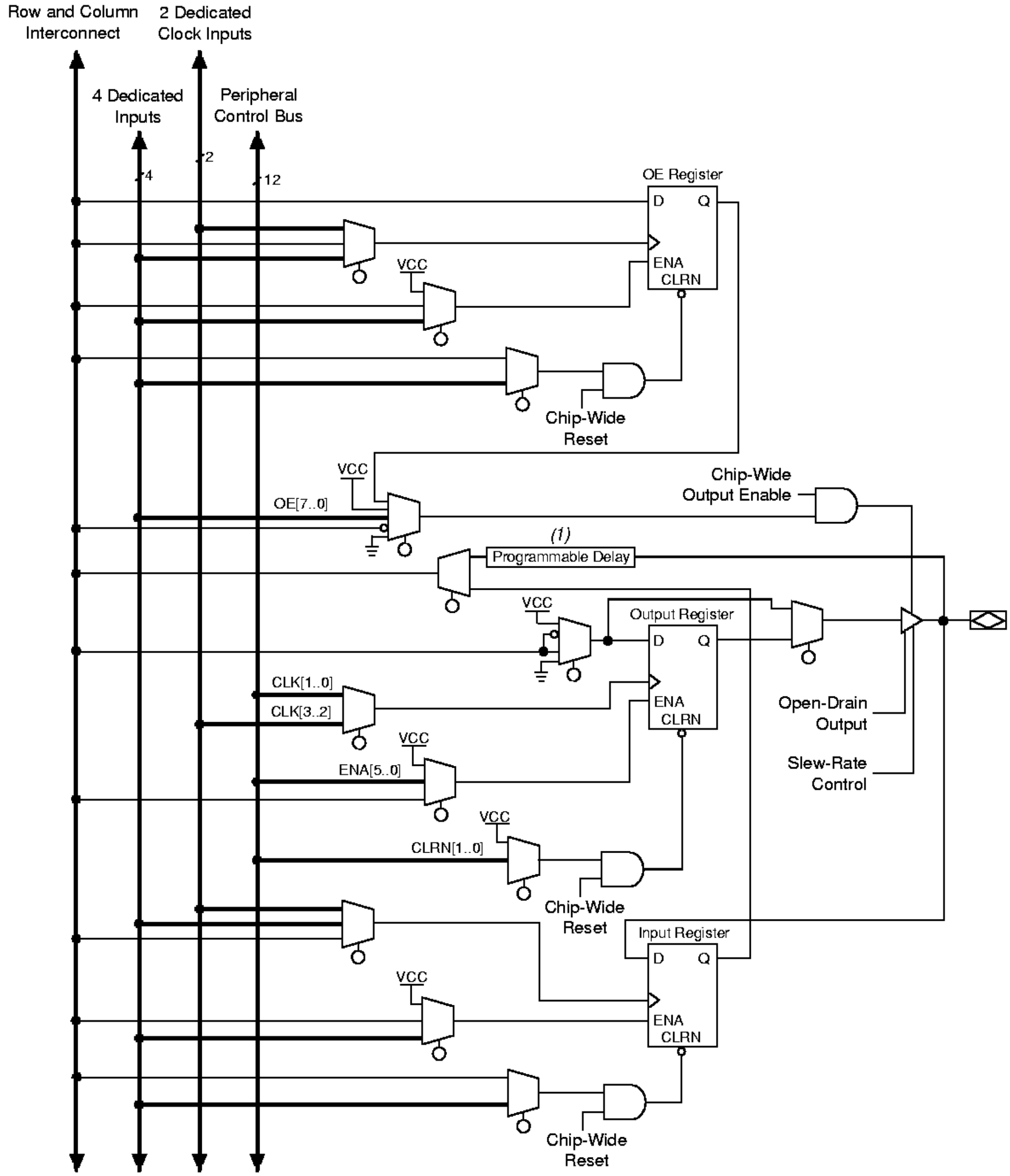
Figure 14. FLEX 10KE Interconnect Resources



I/O Element

An IOE contains a bidirectional I/O buffer and a register that can be used either as an input register for external data that requires a fast setup time, or as an output register for data that requires fast clock-to-output performance. In some cases, using an LE register for an input register will result in a faster setup time than using an IOE register. IOEs can be used as input, output, or bidirectional pins. The Quartus and MAX+PLUS II Compilers use the programmable inversion option to invert signals from the row and column interconnect automatically where appropriate. Figure 15 shows the IOE block diagram.

Figure 15. FLEX 10KE Bidirectional I/O Registers



Note:

(1) Selected FLEX 10KE devices include programmable delay buffers on the input path.

On all FLEX 10KE devices (except EPF10K50E and EPF10K200E), the input path from the I/O pad to the FastTrack Interconnect has a programmable delay element that can be used to guarantee a zero hold time. EPF10K50S and EPF10K200S devices support this feature. Depending on the placement of the IOE relative to what it is driving, the designer may choose to turn on the programmable delay to ensure a zero hold time or turn it off to minimize setup time. This feature is used to reduce setup time for complex pin-to-register paths (e.g., PCI designs).

Each IOE selects the clock, clear, clock enable, and output enable controls from a network of I/O control signals called the peripheral control bus. The peripheral control bus uses high-speed drivers to minimize signal skew across devices and provides up to 12 peripheral control signals that can be allocated as follows:

- Up to eight output enable signals
- Up to six clock enable signals
- Up to two clock signals
- Up to two clear signals

If more than six clock enable or eight output enable signals are required, each IOE on the device can be controlled by clock enable and output enable signals driven by specific LEs. In addition to the two clock signals available on the peripheral control bus, each IOE can use one of two dedicated clock pins. Each peripheral control signal can be driven by any of the dedicated input pins or the first LE of each LAB in a particular row. In addition, a LE in a different row can drive a column interconnect, which causes a row interconnect to drive the peripheral control signal. The chip-wide reset signal resets all IOE registers, overriding any other control signals.

When a dedicated clock pin drives IOE registers, it can be inverted for all IOEs in the device. All IOEs must use the same sense of the clock. For example, if any IOE uses the inverted clock, all IOEs must use the inverted clock and no IOE can use the non-inverted clock. However, LEs can still use the true or complement of the clock on a LAB-by-LAB basis.

The incoming signal may be inverted at the dedicated clock pin and will drive all IOEs. For the true and complement of a clock to be used to drive IOEs, drive it into both global clock pins. One global clock pin will supply the true, and the other will supply the complement.

When the true and complement of a dedicated input drives IOE clocks, two signals on the peripheral control bus are consumed, one for each sense of the clock.

When dedicated inputs drive non-inverted and inverted peripheral clears, clock enables, and output enables, two signals on the peripheral control bus will be used.

Tables 8 and 9 list the sources for each peripheral control signal, and show how the output enable, clock enable, clock, and clear signals share 12 peripheral control signals. The tables also show the rows that can drive global signals.

Table 8. EPF10K30E, EPF10K50E & EPF10K50S Peripheral Bus Sources

Peripheral Control Signal	EPF10K30E	EPF10K50E EPF10K50S
`OE0`	Row A	Row A
`OE1`	Row B	Row B
`OE2`	Row C	Row D
`OE3`	Row D	Row F
`OE4`	Row E	Row H
`OE5`	Row F	Row J
`CLKENA0/CLK0/GLOBAL0`	Row A	Row A
`CLKENA1/OE6/GLOBAL1`	Row B	Row C
`CLKENA2/CLR0`	Row C	Row E
`CLKENA3/OE7/GLOBAL2`	Row D	Row G
`CLKENA4/CLR1`	Row E	Row I
`CLKENA5/CLK1/GLOBAL3`	Row F	Row J

Table 9. EPF10K100β, EPF10K100E, EPF10K130E, EPF10K200E & EPF10K200S Peripheral Bus Sources

Peripheral Control Signal	EPF10K100β EPF10K100E	EPF10K130E	EPF10K200E EPF10K200S
OE0	Row A	Row C	Row G
OE1	Row C	Row E	Row I
OE2	Row E	Row G	Row K
OE3	Row L	Row N	Row R
OE4	Row I	Row K	Row O
OE5	Row K	Row M	Row Q
CLKENA0/CLK0/GLOBAL0	Row F	Row H	Row L
CLKENA1/OE6/GLOBAL1	Row D	Row F	Row J
CLKENA2/CLR0	Row B	Row D	Row H
CLKENA3/OE7/GLOBAL2	Row H	Row J	Row N
CLKENA4/CLR1	Row J	Row L	Row P
CLKENA5/CLK1/GLOBAL3	Row G	Row I	Row M

Signals on the peripheral control bus can also drive the four global signals, referred to as GLOBAL0 through GLOBAL3 in Tables 8 and 9. An internally generated signal can drive a global signal, providing the same low-skew, low-delay characteristics as a signal driven by an input pin. An LE drives the global signal by driving a row line that drives the peripheral bus, which then drives the global signal. This feature is ideal for internally generated clear or clock signals with high fan-out. However, internally driven global signals offer no advantage over the general-purpose interconnect for routing data signals.

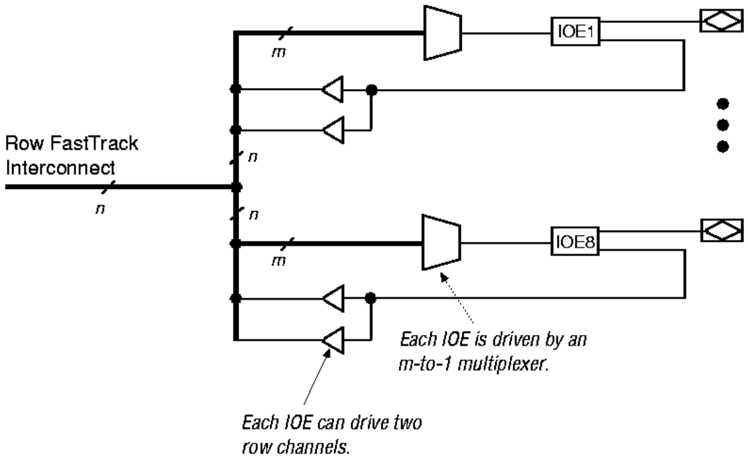
The chip-wide output enable pin is an active-low pin that can be used to tri-state all pins on the device. This option can be set in the MAX+PLUS II software. On EPF10K50E and EPF10K200E devices, the built-in I/O pin pull-up resistors (which are active during configuration) are active when the chip-wide output enable pin is asserted. The registers in the IOE can also be reset by the chip-wide reset pin.

Row-to-IOE Connections

When an IOE is used as an input signal, it can drive two separate row channels. The signal is accessible by all LEs within that row. When an IOE is used as an output, the signal is driven by a multiplexer that selects a signal from the row channels. Up to eight IOEs connect to each side of each row channel (see Figure 16).

Figure 16. FLEX 10KE Row-to-IOE Connections

The values for m and n are provided in Table 10.



Each IOE is driven by an m-to-1 multiplexer.

Each IOE can drive two row channels.

Table 10 lists the FLEX 10KE row-to-IOE interconnect resources.

Table 10. FLEX 10KE Row-to-IOE Interconnect Resources

Device	Channels per Row *(n)*	Row Channels per Pin *(m)*
EPF10K30E	216	27
EPF10K50E EPF10K50S	216	27
EPF10K100B EPF10K100E	312	39
EPF10K130E	312	39
EPF10K200E EPF10K200S	312	39

Column-to-IOE Connections

When an IOE is used as an input, it can drive up to two separate column channels. When an IOE is used as an output, the signal is driven by a multiplexer that selects a signal from the column channels. Two IOEs connect to each side of the column channels. Each IOE can be driven by column channels via a multiplexer. The set of column channels is different for each IOE (see Figure 17).

Figure 17. FLEX 10KE Column-to-IOE Connections

The values for m and n are provided in Table 11.

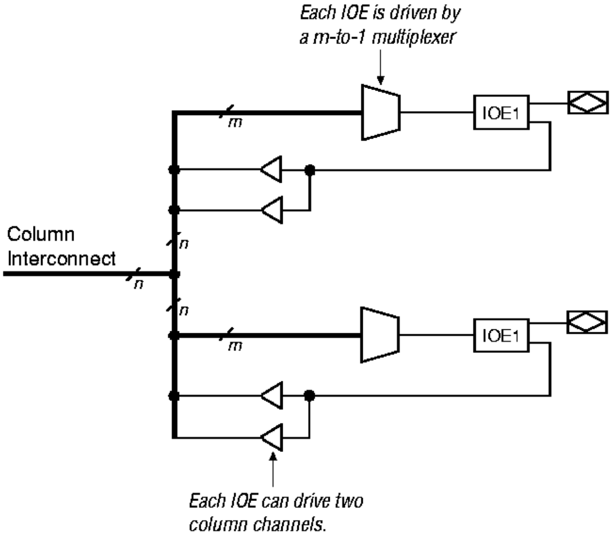


Table 11 lists the FLEX 10KE column-to-IOE interconnect resources.

Table 11. FLEX 10KE Column-to-IOE Interconnect Resources

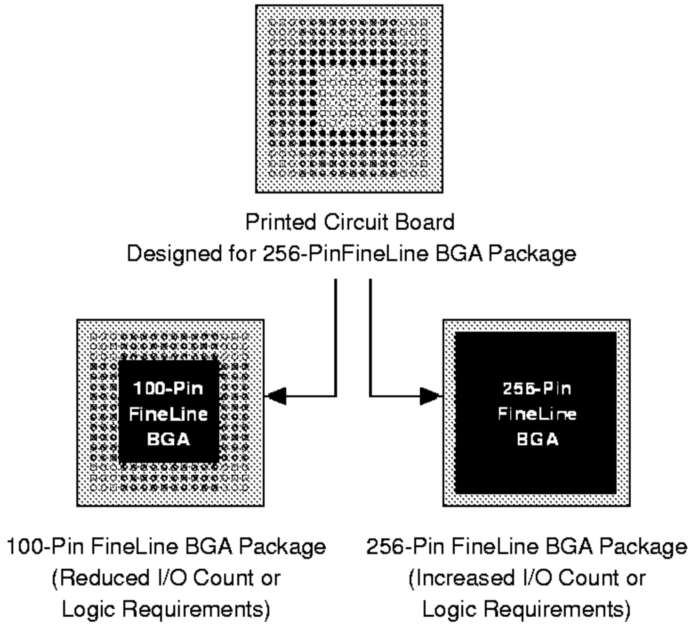
Device	Channels per Column *(n)*	Column Channels per Pin *(m)*
EPF10K30E	24	16
EPF10K50E EPF10K50S	24	16
EPF10K100B EPF10K100E	24	16
EPF10K130E	32	24
EPF10K200E EPF10K200S	48	40

SameFrame Pin-Outs

FLEX 10KE devices support the SameFrame pin-out feature for FineLine BGA packages. The SameFrame pin-out feature is the arrangement of balls on FineLine BGA packages such that the lower-ball-count packages form a subset of the higher-ball-count packages. SameFrame pin-outs provide the flexibility to migrate not only from device to device within the same package, but also from one package to another. A given printed circuit board (PCB) layout can support multiple device density/package combinations. For example, a single board layout can support a range of devices from an EPF10K30E device in a 256-pin FineLine BGA package to an EPF10K200S device in a 672-pin FineLine BGA package.

The Quartus and MAX+PLUS II software provides support to design PCBs with SameFrame pin-out devices. Devices can be defined for present and future use. The Quartus and MAX+PLUS II software generates pin-outs describing how to lay out a board to take advantage of this migration (see Figure 18).

Figure 18. SameFrame Pin-Out Example



ClockLock & ClockBoost Features

To support high-speed designs, FLEX 10KE devices offer optional ClockLock and ClockBoost circuitry containing a phase-locked loop (PLL) that is used to increase design speed and reduce resource usage. The ClockLock circuitry uses a synchronizing PLL that reduces the clock delay and skew within a device. This reduction minimizes clock-to-output and setup times while maintaining zero hold times. The ClockBoost circuitry, which provides a clock multiplier, allows the designer to enhance device area efficiency by resource sharing within the device. The ClockBoost feature allows the designer to distribute a low-speed clock and multiply that clock on-device. Combined, the ClockLock and ClockBoost features provide significant improvements in system performance and bandwidth.

All FLEX 10KE devices, except EPF10K50E devices, support ClockLock and ClockBoost circuitry. EPF10K50S and EPF10K200S devices support this circuitry. Devices that support ClockLock and ClockBoost circuitry are distinguished with an "X" suffix in the ordering code; for instance, the EPF10K200EFC672-1X device supports this circuit.

The ClockLock and ClockBoost features in FLEX 10KE devices are enabled through the Quartus and MAX+PLUS II software. External devices are not required to use these features. The output of the ClockLock and ClockBoost circuits is not available at any of the device pins.

The ClockLock and ClockBoost circuitry locks onto the rising edge of the incoming clock. The circuit output can drive the clock inputs of registers only; the generated clock cannot be gated or inverted.

The dedicated clock pin (`GCLK1`) supplies the clock to the ClockLock and ClockBoost circuitry. When the dedicated clock pin is driving the ClockLock or ClockBoost circuitry, it cannot drive elsewhere in the device.

For designs that require both a multiplied and non-multiplied clock, the clock trace on the board can be connected to the `GCLK1` pin. In the MAX+PLUS II software, the `GCLK1` pin can feed both the ClockLock and ClockBoost circuitry in the FLEX 10KE device. However, when both circuits are used, the other clock pin cannot be used.

ClockLock & ClockBoost Timing Parameters

For the ClockLock and ClockBoost circuitry to function properly, the incoming clock must meet certain requirements. If these specifications are not met, the circuitry may not lock onto the incoming clock, which generates an erroneous clock within the device. The clock generated by the ClockLock and ClockBoost circuitry must also meet certain specifications. If the incoming clock meets these requirements during configuration, the ClockLock and ClockBoost circuitry will lock onto the clock during configuration. The circuit will be ready for use immediately after configuration. Figure 19 shows the incoming and generated clock specifications.

Figure 19. Specifications for the Incoming & Generated Clocks

The t_I parameter refers to the nominal input clock period; the t_O parameter refers to the nominal output clock period.

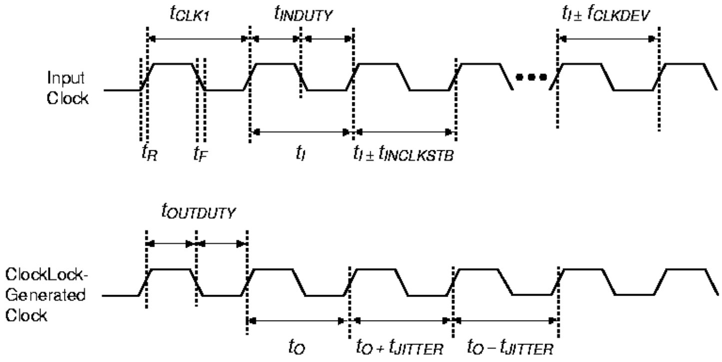


Table 12 summarizes the ClockLock and ClockBoost parameters.

Table 12. ClockLock & ClockBoost Parameters

Symbol	Parameter	Min	Typ	Max	Unit
t_R	Input rise time			5	ns
t_F	Input fall time			5	ns
t_{INDUTY}	Input duty cycle	40		60	%
f_{CLK1}	Input clock frequency (ClockBoost clock multiplication factor equals 1)	30		140	MHz
f_{CLK2}	Input clock frequency (ClockBoost clock multiplication factor equals 2)	16		70	MHz
f_{CLKDEV}	Input deviation from user specification in the MAX+PLUS II software *(1)*			25,000 *(2)*	PPM
$t_{INCLKSTB}$	Input clock stability (measured between adjacent clocks)			100	ps
t_{LOCK}	Time required for ClockLock or ClockBoost to acquire lock *(3)*			10	µs
t_{JITTER}	Jitter on ClockLock or ClockBoost-generated clock *(4)*			250 *(4)*	ps
$t_{OUTDUTY}$	Duty cycle for ClockLock or ClockBoost-generated clock	40	50	60	%

Notes:

(1) To implement the ClockLock and ClockBoost circuitry with the MAX+PLUS II software, designers must specify the input frequency. The MAX+PLUS II software tunes the PLL in the ClockLock and ClockBoost circuitry to this frequency. The f_{CLKDEV} parameter specifies how much the incoming clock can differ from the specified frequency during device operation. Simulation does not reflect this parameter.

(2) Twenty-five thousand parts per million (PPM) equates to 2.5% of input clock period.

(3) During device configuration, the ClockLock and ClockBoost circuitry is configured before the rest of the device. If the incoming clock is supplied during configuration, the ClockLock and ClockBoost circuitry locks during configuration because the t_{LOCK} value is less than the time required for configuration.

(4) The t_{JITTER} specification is measured under long-term observation. The maximum value for t_{JITTER} is 200 ps if $t_{INCLKSTB}$ is lower than 50 ps.

I/O Configuration

This section discusses the peripheral component interconnect (PCI) pull-up clamping diode option, slew-rate control, open-drain output option, and MultiVolt I/O interface for FLEX 10KE devices. The PCI pull-up clamping diode, slew-rate control, and open-drain output options are controlled pin-by-pin via MAX+PLUS II logic options. The MultiVolt I/O interface is controlled by connecting V_{CCIO} to a different voltage than V_{CCINT}. Its effect can be simulated in the MAX+PLUS II software via the **Global Project Device Options** dialog box (Assign menu).

PCI Pull-Up Clamping Diode Option

FLEX 10KE devices have a pull-up clamping diode on every I/O, dedicated input, and dedicated clock pin. PCI clamping diodes clamp the signal to the V_{CCIO} value and are required for 3.3-V PCI compliance. Clamping diodes can also be used to limit overshoot in other systems.

Clamping diodes are controlled on a pin-by-pin basis. When V_{CCIO} is 3.3 V, a pin that has the clamping diode option turned on can be driven by a 2.5-V or 3.3-V signal, but not a 5.0-V signal. When V_{CCIO} is 2.5 V, a pin that has the clamping diode option turned on can be driven by a 2.5-V signal, but not a 3.3-V or 5.0-V signal. Additionally, a clamping diode can be activated for a subset of pins, which would allow a device to bridge between a 3.3-V PCI bus and a 5.0-V device.

Slew-Rate Control

The output buffer in each IOE has an adjustable output slew rate that can be configured for low-noise or high-speed performance. A slower slew rate reduces system noise and adds a maximum delay of 4.3 ns. The fast slew rate should be used for speed-critical outputs in systems that are adequately protected against noise. Designers can specify the slew rate pin-by-pin or assign a default slew rate to all pins on a device-wide basis. The slow slew rate setting affects only the falling and rising edges of the output.

Open-Drain Output Option

FLEX 10KE devices provide an optional open-drain output (electrically equivalent to open-collector output) for each I/O pin. This open-drain output enables the device to provide system-level control signals (e.g., interrupt and write enable signals) that can be asserted by any of several devices. It can also provide an additional wired-`OR` plane.

MultiVolt I/O Interface

The FLEX 10KE device architecture supports the MultiVolt I/O interface feature, which allows FLEX 10KE devices in all packages to interface with systems of differing supply voltages. These devices have one set of V_{CC} pins for internal operation and input buffers (`VCCINT`), and another set for I/O output drivers (`VCCIO`).

The `VCCINT` pins must always be connected to a 2.5-V power supply. With a 2.5-V V_{CCINT} level, input voltages are compatible with 2.5-V, 3.3-V, and 5.0-V inputs. The `VCCIO` pins can be connected to either a 2.5-V or 3.3-V power supply, depending on the output requirements. When the `VCCIO` pins are connected to a 2.5-V power supply, the output levels are compatible with 2.5-V systems. When the `VCCIO` pins are connected to a 3.3-V power supply, the output high is at 3.3 V and is therefore compatible with 3.3-V or 5.0-V systems. Devices operating with V_{CCIO} levels higher than 3.0 V achieve a faster timing delay of t_{OD2} instead of t_{OD1}.

Table 13 summarizes FLEX 10KE MultiVolt I/O support.

Table 13. FLEX 10KE MultiVolt I/O Support

V_{CCIO} (V)	Input Signal (V)			Output Signal (V)		
	2.5	3.3	5.0	2.5	3.3	5.0
2.5	✓	✓ *(1)*	✓ *(1)*	✓		
3.3	✓	✓	✓ *(1)*	✓ *(2)*	✓	✓

Notes:

(1) The PCI clamping diode must be disabled to drive an input with voltages higher than V_{CCIO}.

(2) When V_{CCIO} = 3.3 V, a FLEX 10KE device can drive a 2.5-V device that has 3.3-V tolerant inputs.

Open-drain output pins on FLEX 10KE devices (with a pull-up resistor to the 5.0-V supply) can drive 5.0-V CMOS input pins that require a V_{IH} of 3.5 V. When the open-drain pin is active, it will drive low. When the pin is inactive, the trace will be pulled up to 5.0 V by the resistor. The open-drain pin will only drive low or tri-state; it will never drive high. The rise time is dependent on the value of the pull-up resistor and load impedance. The I_{OL} current specification should be considered when selecting a pull-up resistor.

Power Sequencing & Hot-Socketing

Because FLEX 10KE devices can be used in a mixed-voltage environment, they have been designed specifically to tolerate any possible power-up sequence. The V_{CCIO} and V_{CCINT} power planes can be powered in any order.

Signals can be driven into FLEX 10KE devices before and during power up without damaging the device. Additionally, FLEX 10KE devices do not drive out during power up. Once operating conditions are reached, FLEX 10KE devices operate as specified by the user.

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All FLEX 10KE devices provide JTAG BST circuitry that complies with the IEEE Std. 1149.1-1990 specification. FLEX 10KE devices can also be configured using the JTAG pins through the BitBlaster or ByteBlasterMV download cable, or via hardware that uses the Jam™ programming and test language. JTAG boundary-scan testing can be performed before or after configuration, but not during configuration. FLEX 10KE devices support the JTAG instructions shown in Table 14.

Table 14. FLEX 10KE JTAG Instructions

JTAG Instruction	Description
SAMPLE/PRELOAD	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation, and permits an initial data pattern to be output at the device pins.
EXTEST	Allows the external circuitry and board-level interconnections to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	Places the 1-bit bypass register between the `TDI` and `TDO` pins, which allows the BST data to pass synchronously through a selected device to adjacent devices during normal device operation.
USERCODE	Selects the user electronic signature (USERCODE) register and places it between the `TDI` and `TDO` pins, allowing the USERCODE to be serially shifted out of `TDO`.
IDCODE	Selects the IDCODE register and places it between `TDI` and `TDO`, allowing the IDCODE to be serially shifted out of `TDO`.
ICR Instructions	These instructions are used when configuring a FLEX 10KE device via JTAG ports with a BitBlaster or ByteBlasterMV download cable, or using a Jam File (**.jam**) or Jam Byte-Code File (**.jbc**) via an embedded processor.

The instruction register length of FLEX 10KE devices is 10 bits. The USERCODE register length in FLEX 10KE devices is 32 bits; 7 bits are determined by the user, and 25 bits are pre-determined. Tables 15 and 16 show the boundary-scan register length and device IDCODE information for FLEX 10KE devices.

Table 15. FLEX 10KE Boundary-Scan Register Length

Device	Boundary-Scan Register Length
EPF10K30E	690
EPF10K50E EPF10K50S	798
EPF10K100B	873
EPF10K100E	1,050
EPF10K130E	1,308
EPF10K200E EPF10K200S	1,446

Table 16. 32-Bit IDCODE for FLEX 10KE Devices *Note (1)*

Device	IDCODE (32 Bits)			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer's Identity (11 Bits)	1 (1 Bit) *(2)*
EPF10K30E	`0001`	`0001 0000 0011 0000`	`00001101110`	`1`
EPF10K50E EPF10K50S	`0001`	`0001 0000 0101 0000`	`00001101110`	`1`
EPF10K100B	`0001`	`0000 0001 0000 0000`	`00001101110`	`1`
EPF10K100E	`0010`	`0000 0001 0000 0000`	`00001101110`	`1`
EPF10K130E	`0001`	`0000 0001 0011 0000`	`00001101110`	`1`
EPF10K200E EPF10K200S	`0001`	`0000 0010 0000 0000`	`00001101110`	`1`

Notes:

(1) The most significant bit (MSB) is on the left.

(2) The least significant bit (LSB) for all JTAG IDCODEs is `1`.

FLEX 10KE devices include weak pull-up resistors on the JTAG pins.

For more information, see the following documents:

- *Application Note 39 (IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices)*
- *BitBlaster Serial Download Cable Data Sheet*
- *ByteBlasterMV Parallel Port Download Cable Data Sheet*
- *Jam Programming & Test Language Specification*

Figure 20 shows the timing requirements for the JTAG signals.

Figure 20. FLEX 10KE JTAG Waveforms

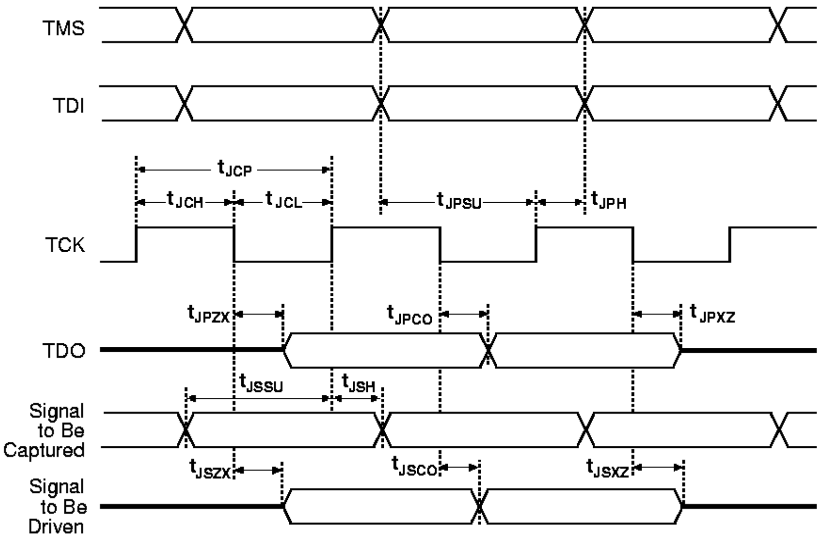


Table 17 shows the timing parameters and values for FLEX 10KE devices.

Table 17. FLEX 10KE JTAG Timing Parameters & Values

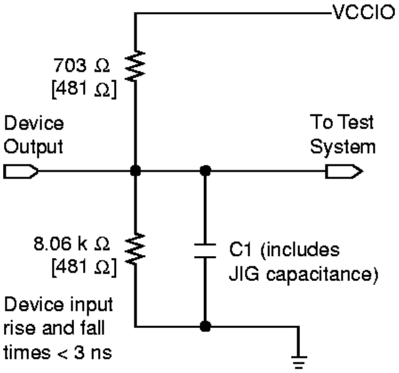
Symbol	Parameter	Min	Max	Unit
t_{JCP}	`TCK` clock period	100		ns
t_{JCH}	`TCK` clock high time	50		ns
t_{JCL}	`TCK` clock low time	50		ns
t_{JPSU}	JTAG port setup time	20		ns
t_{JPH}	JTAG port hold time	45		ns
t_{JPCO}	JTAG port clock to output		25	ns
t_{JPZX}	JTAG port high impedance to valid output		25	ns
t_{JPXZ}	JTAG port valid output to high impedance		25	ns
t_{JSSU}	Capture register setup time	20		ns
t_{JSH}	Capture register hold time	45		ns
t_{JSCO}	Update register clock to output		35	ns
t_{JSZX}	Update register high impedance to valid output		35	ns
t_{JSXZ}	Update register valid output to high impedance		35	ns

Generic Testing

Each FLEX 10KE device is functionally tested. Complete testing of each configurable static random access memory (SRAM) bit and all logic functionality ensures 100% yield. AC test measurements for FLEX 10KE devices are made under conditions equivalent to those shown in Figure 21. Multiple test patterns can be used to configure devices during all stages of the production flow.

Figure 21. FLEX 10KE AC Test Conditions

Power supply transients can affect AC measurements. Simultaneous transitions of multiple outputs should be avoided for accurate measurement. Threshold tests must not be performed under AC conditions. Large-amplitude, fast-ground-current transients normally occur as the device outputs discharge the load capacitances. When these transients flow through the parasitic inductance between the device ground pin and the test system ground, significant reductions in observable noise immunity can result. Numbers in brackets are for 2.5-V devices or outputs. Numbers without brackets are for 3.3-V devices or outputs.



Operating Conditions

Tables 18 through 21 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 2.5-V FLEX 10KE devices.

Table 18. FLEX 10KE 2.5-V Device Absolute Maximum Ratings *Note (1)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage	With respect to ground *(2)*	–0.5	3.6	V
V_{CCIO}			–0.5	4.6	V
V_I	DC input voltage		–2.0	5.75	V
I_{OUT}	DC output current, per pin		–25	25	mA
T_{STG}	Storage temperature	No bias	–65	150	° C
T_{AMB}	Ambient temperature	Under bias	–65	135	° C
T_J	Junction temperature	PQFP, TQFP, and BGA packages, under bias		135	° C
		Ceramic PGA packages, under bias		150	° C

Table 19. FLEX 10KE 2.5-V Device Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	*(3)*, *(4)*	2.30 (2.30)	2.70 (2.70)	V
V_{CCIO}	Supply voltage for output buffers, 3.3-V operation	*(3)*, *(4)*	3.00 (3.00)	3.60 (3.60)	V
	Supply voltage for output buffers, 2.5-V operation	*(3)*, *(4)*	2.30 (2.30)	2.70 (2.70)	V
V_I	Input voltage	*(5)*	–0.5	5.75	V
V_O	Output voltage		0	V_{CCIO}	V
T_A	Ambient temperature	For commercial use	0	70	° C
		For industrial use	–40	85	° C
T_J	Operating temperature	For commercial use	0	85	° C
		For industrial use	–40	100	° C
t_R	Input rise time			40	ns
t_F	Input fall time			40	ns

Table 20. FLEX 10KE 2.5-V Device DC Operating Conditions *Notes (6), (7)*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	High-level input voltage		1.7, 0.5 × V_{CCIO} *(8)*		5.75	V
V_{IL}	Low-level input voltage		–0.5		0.8, 0.3 × V_{CCIO} *(8)*	V
V_{OH}	3.3-V high-level TTL output voltage	I_{OH} = –8 mA DC, V_{CCIO} = 3.00 V *(9)*	2.4			V
	3.3-V high-level CMOS output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 3.00 V *(9)*	V_{CCIO} – 0.2			V
	3.3-V high-level PCI output voltage	I_{OH} = –0.5 mA DC, V_{CCIO} = 3.00 to 3.60 V *(9)*	0.9 × V_{CCIO}			V
	2.5-V high-level output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 2.30 V *(9)*	2.1			V
		I_{OH} = –1 mA DC, V_{CCIO} = 2.30 V *(9)*	2.0			V
		I_{OH} = –2 mA DC, V_{CCIO} = 2.30 V *(9)*	1.7			V
V_{OL}	3.3-V low-level TTL output voltage	I_{OL} = 12 mA DC, V_{CCIO} = 3.00 V *(9)*			0.45	V
	3.3-V low-level CMOS output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 3.00 V *(10)*			0.2	V
	3.3-V low-level PCI output voltage	I_{OL} = 1.5 mA DC, V_{CCIO} = 3.00 to 3.60 V *(10)*			0.1 × V_{CCIO}	V
	2.5-V low-level output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 2.30 V *(10)*			0.2	V
		I_{OL} = 1 mA DC, V_{CCIO} = 2.30 V *(10)*			0.4	V
		I_{OL} = 2 mA DC, V_{CCIO} = 2.30 V *(10)*			0.7	V
I_I	Input pin leakage current	V_I = 5.3 to –0.3 V	–10		10	μA
I_{OZ}	Tri-stated I/O pin leakage current	V_O = 5.3 to –0.3 V	–10		10	μA
I_{CC0}	V_{CC} supply current (standby)	V_I = ground, no load, no toggling inputs		5		mA
		V_I = ground, no load, no toggling inputs *(11)*		10		mA
R_{CONF}	Value of I/O pin pull-up resistor before and during configuration	V_{CCIO} = 3.0 V *(12)*	20		50	kΩ
		V_{CCIO} = 2.3 V *(12)*	30		80	kΩ

Table 21. FLEX 10KE Device Capacitance *Note (13)*

Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Input capacitance	V_{IN} = 0 V, f = 1.0 MHz		10	pF
C_{INCLK}	Input capacitance on dedicated clock pin	V_{IN} = 0 V, f = 1.0 MHz		12	pF
C_{OUT}	Output capacitance	V_{OUT} = 0 V, f = 1.0 MHz		10	pF

Notes to tables:

(1) See the *Operating Requirements for Altera Devices Data Sheet* in this data book.
(2) Minimum DC input voltage is –0.5 V. During transitions, the inputs may undershoot to –2.0 V or overshoot to 5.75 V for input currents less than 100 mA and periods shorter than 20 ns.
(3) Numbers in parentheses are for industrial-temperature-range devices.
(4) Maximum V_{CC} rise time is 100 ms, and V_{CC} must rise monotonically.
(5) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT} and V_{CCIO} are powered.
(6) Typical values are for T_A = 25° C, V_{CCINT} = 2.5 V, and V_{CCIO} = 2.5 V or 3.3 V.
(7) These values are specified under the FLEX 10KE Recommended Operating Conditions shown in Table 19 on page 47.
(8) The FLEX 10KE input buffers are compatible with 2.5-V, 3.3-V (LVTTL and LVCMOS), and 5.0-V TTL and CMOS signals. Additionally, the input buffers are 3.3-V PCI compliant when V_{CCIO} and V_{CCINT} meet the relationship shown in Figure 22.
(9) The I_{OH} parameter refers to high-level TTL, PCI, or CMOS output current.
(10) The I_{OL} parameter refers to low-level TTL, PCI, or CMOS output current. This parameter applies to open-drain pins as well as output pins.
(11) This parameter applies to -1 speed grade commercial temperature devices and -2 speed grade industrial temperature devices.
(12) Pin pull-up resistance values will be lower if the pin is driven higher than V_{CCIO} by an external source.
(13) Capacitance is sample-tested only.

Figure 22 shows the required relationship between V_{CCIO} and V_{CCINT} for 3.3-V PCI compliance.

Figure 22. Relationship between V_{CCIO} & V_{CCINT} for 3.3-V PCI Compliance

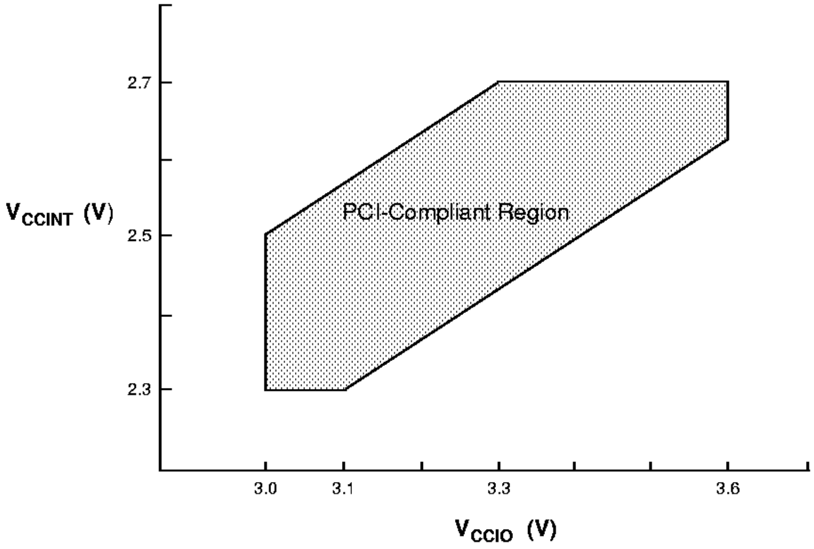
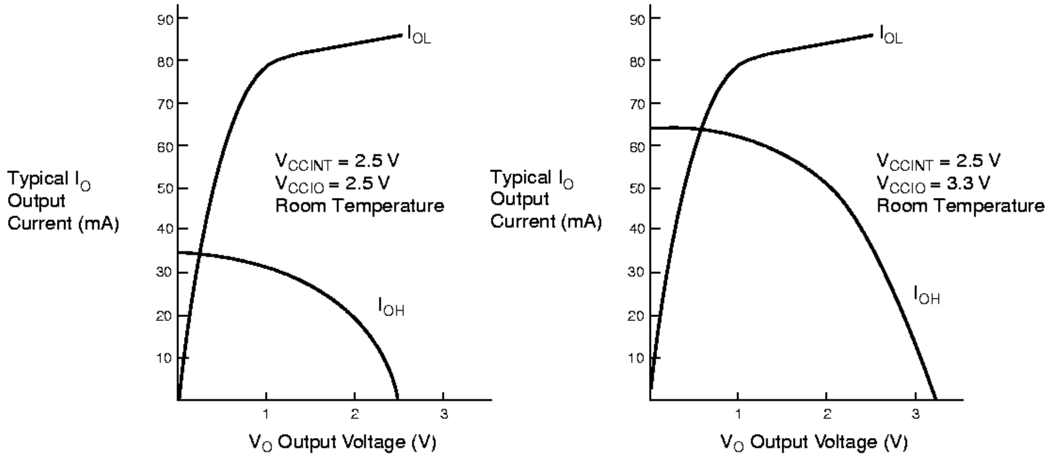


Figure 23 shows the typical output drive characteristics of FLEX 10KE devices with 3.3-V and 2.5-V V_{CCIO}. The output driver is compliant to the 3.3-V ***PCI Local Bus Specification, Revision 2.2*** (when `VCCIO` pins are connected to 3.3 V). FLEX 10KE devices with a -1 speed grade also comply with the drive strength requirements of the ***PCI Local Bus Specification, Revision 2.2*** (when `VCCINT` pins are powered with a minimum supply of 2.375 V, and `VCCIO` pins are connected to 3.3 V). Therefore, these devices can be used in open 5.0-V PCI systems.

Figure 23. Output Drive Characteristics of FLEX 10KE Devices



Timing Model

The continuous, high-performance FastTrack Interconnect routing resources ensure predictable performance and accurate simulation and timing analysis. This predictable performance contrasts with that of FPGAs, which use a segmented connection scheme and therefore have unpredictable performance.

Device performance can be estimated by following the signal path from a source, through the interconnect, to the destination. For example, the registered performance between two LEs on the same row can be calculated by adding the following parameters:

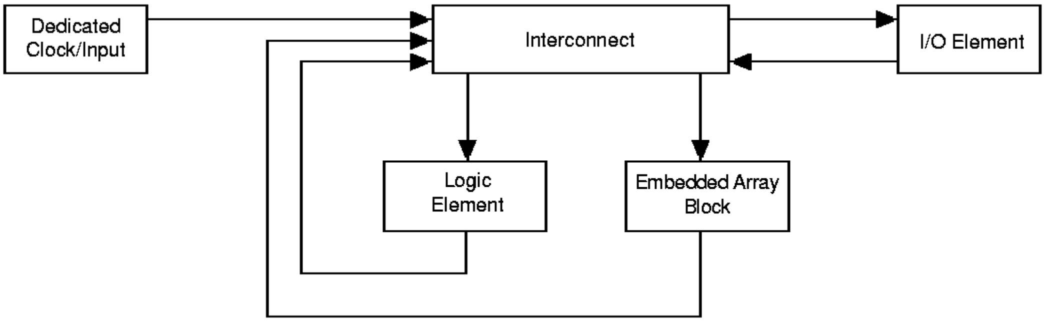
- LE register clock-to-output delay (t_{CO})
- Interconnect delay ($t_{SAMEROW}$)
- LE look-up table delay (t_{LUT})
- LE register setup time (t_{SU})

The routing delay depends on the placement of the source and destination LEs. A more complex registered path may involve multiple combinatorial LEs between the source and destination LEs.

Timing simulation and delay prediction are available with the MAX+PLUS II Simulator and Timing Analyzer, or with industry-standard EDA tools. The Simulator offers both pre-synthesis functional simulation to evaluate logic design accuracy and post-synthesis timing simulation with 0.1-ns resolution. The Timing Analyzer provides point-to-point timing delay information, setup and hold time analysis, and device-wide performance analysis.

Figure 24 shows the overall timing model, which maps the possible paths to and from the various elements of the FLEX 10KE device.

Figure 24. FLEX 10KE Device Timing Model



Figures 25 through 28 show the delays that correspond to various paths and functions within the LE, IOE, EAB, and bidirectional timing models.

Figure 25. FLEX 10KE Device LE Timing Model

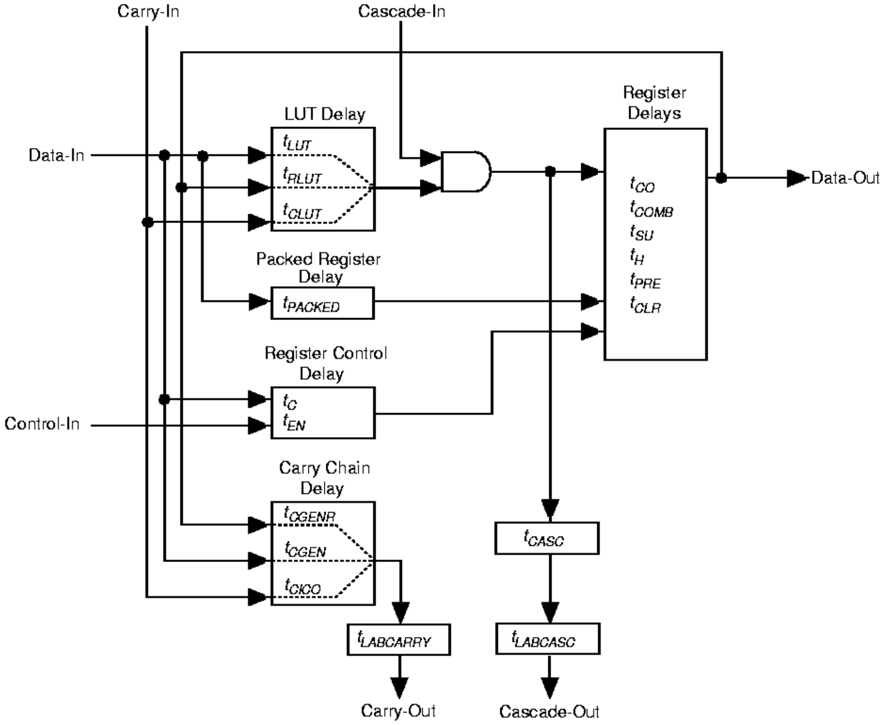


Figure 26. FLEX 10KE Device IOE Timing Model

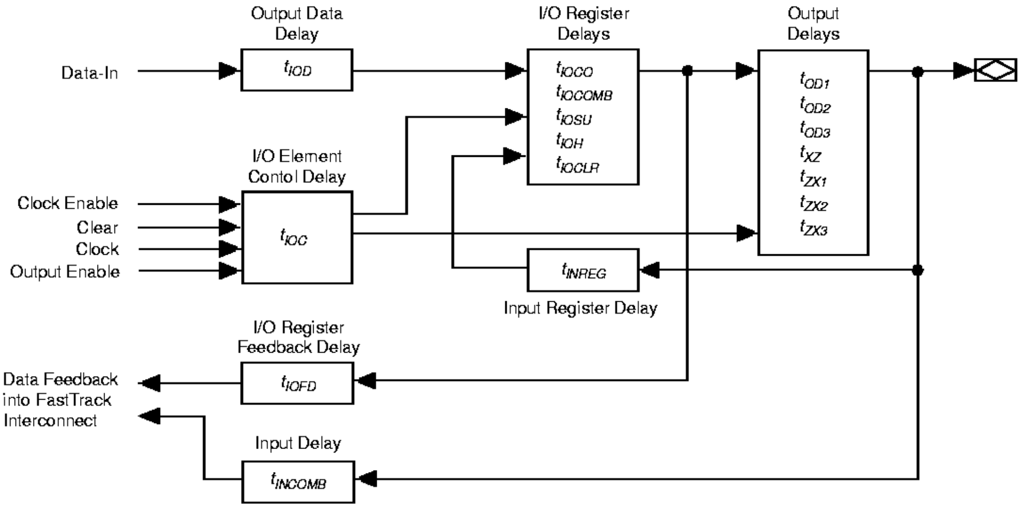


Figure 27. FLEX 10KE Device EAB Timing Model

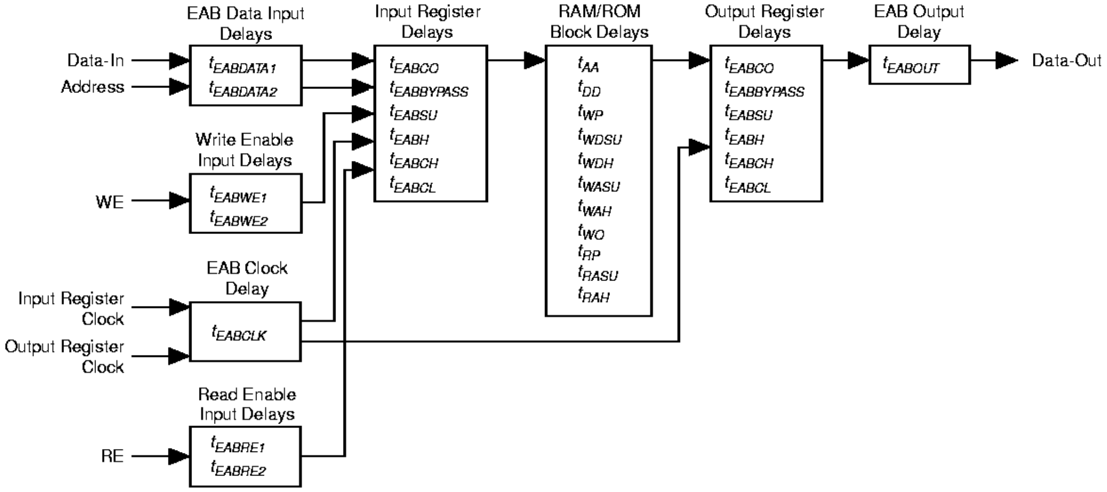
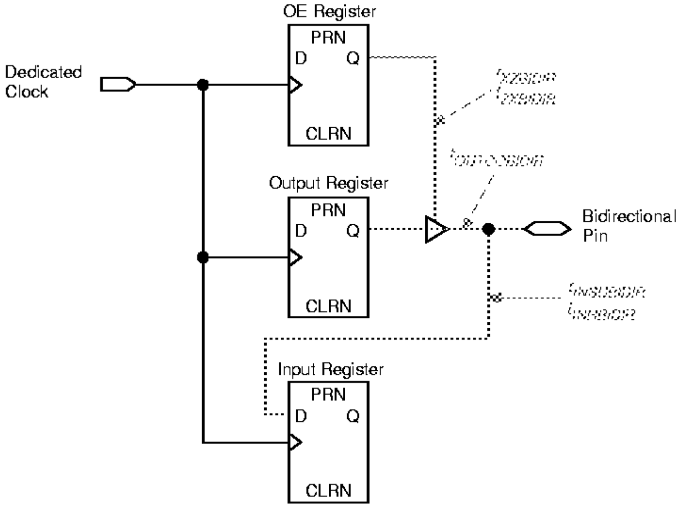


Figure 28. Synchronous Bidirectional Pin External Timing Model



Tables 22 through 26 describe the FLEX 10KE device internal timing parameters. Tables 27 through 29 describe the FLEX 10KE external timing parameters and their symbols. Detailed timing information for these devices will be released as it is available.

Table 22. LE Timing Microparameters (Part 1 of 2) *Note (1)*

Symbol	Parameter	Conditions
t_{LUT}	LUT delay for data-in	
t_{CLUT}	LUT delay for carry-in	
t_{RLUT}	LUT delay for LE register feedback	
t_{PACKED}	Data-in to packed register delay	
t_{EN}	LE register enable delay	
t_{CICO}	Carry-in to carry-out delay	
t_{CGEN}	Data-in to carry-out delay	
t_{CGENR}	LE register feedback to carry-out delay	
t_{CASC}	Cascade-in to cascade-out delay	
t_C	LE register control signal delay	
t_{CO}	LE register clock-to-output delay	
t_{COMB}	Combinatorial delay	
t_{SU}	LE register setup time for data and enable signals before clock; LE register recovery time after asynchronous clear, preset, or load	
t_H	LE register hold time for data and enable signals after clock	
t_{PRE}	LE register preset delay	

Table 22. LE Timing Microparameters (Part 2 of 2) *Note (1)*

Symbol	Parameter	Conditions
t_{CLR}	LE register clear delay	
t_{CH}	Minimum clock high time from clock pin	
t_{CL}	Minimum clock low time from clock pin	

Table 23. IOE Timing Microparameters *Note (1)*

Symbol	Parameter	Conditions
t_{IOD}	IOE data delay	
t_{IOC}	IOE register control signal delay	
t_{IOCO}	IOE register clock-to-output delay	
t_{IOCOMB}	IOE combinatorial delay	
t_{IOSU}	IOE register setup time for data and enable signals before clock; IOE register recovery time after asynchronous clear	
t_{IOH}	IOE register hold time for data and enable signals after clock	
t_{IOCLR}	IOE register clear time	
t_{OD1}	Output buffer and pad delay, slow slew rate = off, V_{CCIO} = V_{CCINT}	C1 = 35 pF *(2)*
t_{OD2}	Output buffer and pad delay, slow slew rate = off, V_{CCIO} = low voltage	C1 = 35 pF *(3)*
t_{OD3}	Output buffer and pad delay, slow slew rate = on	C1 = 35 pF *(4)*
t_{XZ}	IOE output buffer disable delay	
t_{ZX1}	IOE output buffer enable delay, slow slew rate = off, V_{CCIO} = V_{CCINT}	C1 = 35 pF *(2)*
t_{ZX2}	IOE output buffer enable delay, slow slew rate = off, V_{CCIO} = low voltage	C1 = 35 pF *(3)*
t_{ZX3}	IOE output buffer enable delay, slow slew rate = on	C1 = 35 pF *(4)*
t_{INREG}	IOE input pad and buffer to IOE register delay	
t_{IOFD}	IOE register feedback delay	
t_{INCOMB}	IOE input pad and buffer to FastTrack Interconnect delay	

Table 24. EAB Timing Microparameters *Note (1)*

Symbol	Parameter	Conditions
$t_{EABDATA1}$	Data or address delay to EAB for combinatorial input	
$t_{EABDATA2}$	Data or address delay to EAB for registered input	
t_{EABWE1}	Write enable delay to EAB for combinatorial input	
t_{EABWE2}	Write enable delay to EAB for registered input	
t_{EABRE1}	Read enable delay to EAB for combinatorial input	
t_{EABRE2}	Read enable delay to EAB for registered input	
t_{EABCLK}	EAB register clock delay	
t_{EABCO}	EAB register clock-to-output delay	
$t_{EABBYPASS}$	Bypass register delay	
t_{EABSU}	EAB register setup time before clock	
t_{EABH}	EAB register hold time after clock	
t_{EABCLR}	EAB register asynchronous clear time to output delay	
t_{EABCH}	Clock high time	
t_{EABCL}	Clock low time	
t_{AA}	Address access delay (including the read enable to output delay)	
t_{WP}	Write pulse width	
t_{RP}	Read pulse width	
t_{WDSU}	Data setup time before falling edge of write pulse	*(5)*
t_{WDH}	Data hold time after falling edge of write pulse	*(5)*
t_{WASU}	Address setup time before rising edge of write pulse	*(5)*
t_{WAH}	Address hold time after falling edge of write pulse	*(5)*
t_{WASU}	Address setup time before rising edge of write pulse	
t_{WAH}	Address hold time after falling edge of write pulse	
t_{RASU}	Address setup time before rising edge of read pulse	
t_{RAH}	Address hold time after falling edge of read pulse	
t_{WO}	Write enable to data output valid delay	
t_{DD}	Data-in to data-out valid delay	
t_{EABOUT}	Data-out delay	

Table 25. EAβ Timing Macroparameters *Notes (1), (6)*

Symbol	Parameter	Conditions
t_{EABAA}	EAB address access delay	
$t_{EABRCCOMB}$	EAB asynchronous read cycle time	
$t_{EABRCREG}$	EAB synchronous read cycle time	
t_{EABWP}	EAB write pulse width	
$t_{EABWCCOMB}$	EAB asynchronous write cycle time	
$t_{EABWCREG}$	EAB synchronous write cycle time	
t_{EABDD}	EAB data-in to data-out valid delay	
$t_{EABDATACO}$	EAB clock-to-output delay when using output registers	
$t_{EABDATASU}$	EAB data/address setup time before clock when using input register	
$t_{EABDATAH}$	EAB data/address hold time after clock when using input register	
$t_{EABWESU}$	EAB `WE` setup time before clock when using input register	
$t_{EABWESH}$	EAB `WE` hold time after clock when using input register	
$t_{EABWDSU}$	EAB data setup time before falling edge of write pulse when not using input registers	
t_{EABWDH}	EAB data hold time after falling edge of write pulse when not using input registers	
$t_{EABWASU}$	EAB address setup time before rising edge of write pulse when not using input registers	
t_{EABWAH}	EAB address hold time after falling edge of write pulse when not using input registers	
t_{EABWO}	EAB write enable to data output valid delay	

Table 26. Interconnect Timing Microparameters *Note (1)*

Symbol	Parameter	Conditions
$t_{SAMELAB}$	Routing delay for an LE driving another LE in the same LAB	
$t_{SAMEROW}$	Routing delay for a row IOE, LE, or EAB driving a row IOE, LE, or EAB in the same row	*(7)*
$t_{SAMECOLUMN}$	Routing delay for an LE driving an IOE in the same column	*(7)*
$t_{DIFFROW}$	Routing delay for a column IOE, LE, or EAB driving an LE or EAB in a different row	*(7)*
$t_{TWOROWS}$	Routing delay for a row IOE or EAB driving an LE or EAB in a different row	*(7)*
$t_{LEPERIPH}$	Routing delay for an LE driving a control signal of an IOE via the peripheral control bus	*(7)*
$t_{LABCARRY}$	Routing delay for the carry-out signal of an LE driving the carry-in signal of a different LE in a different LAB	
$t_{LABCASC}$	Routing delay for the cascade-out signal of an LE driving the cascade-in signal of a different LE in a different LAB	
$t_{DIN2IOE}$	Delay from dedicated input pin to IOE control input	*(7)*
t_{DIN2LE}	Delay from dedicated input pin to LE or EAB control input	*(7)*
$t_{DCLK2IOE}$	Delay from dedicated clock pin to IOE clock	*(7)*
$t_{DCLK2LE}$	Delay from dedicated clock pin to LE or EAB clock	*(7)*
$t_{DIN2DATA}$	Delay from dedicated input or clock to LE or EAB data	*(7)*

Table 27. External Reference Timing Parameters *Note (8)*

Symbol	Parameter	Conditions
t_{DRR}	Register-to-register delay via four LEs, three row interconnects, and four local interconnects	*(9)*

Table 28. External Timing Parameters

Symbol	Parameter	Conditions
t_{DRR}	Register-to-register delay via four LEs, three row interconnects, and four local interconnects	*(11)*
t_{INSU}	Setup time with global clock at IOE register	
t_{INH}	Hold time with global clock at IOE register	
t_{OUTCO}	Clock-to-output delay with global clock at IOE register	
t_{PCISU}	Setup time with global clock for registers used in PCI designs	*(12)*
t_{PCIH}	Hold time with global clock for registers used in PCI designs	*(12)*
t_{PCICO}	Clock-to-output delay with global clock for registers used in PCI designs	*(12)*

Table 29. External Bidirectional Timing Parameters *Note (10)*

Symbol	Parameter	Condition
$t_{INSUBIDIR}$	Setup time for bidirectional pins with global clock at same-row or same-column LE register	
$t_{INHBIDIR}$	Hold time for bidirectional pins with global clock at same-row or same-column LE register	
$t_{OUTCOBIDIR}$	Clock-to-output delay for bidirectional pins with global clock at IOE register	CI = 35 pF
$t_{XZBIDIR}$	Synchronous IOE output buffer disable delay	CI = 35 pF
$t_{ZXBIDIR}$	Synchronous IOE output buffer enable delay, slow slew rate = off	CI = 35 pF

Notes to tables:

(1) Microparameters are timing delays contributed by individual architectural elements. These parameters cannot be measured explicitly.
(2) Operating conditions: V_{CCIO} = 3.3 V ± 10% for commercial or industrial use in FLEX 10KE devices
(3) Operating conditions: V_{CCIO} = 2.5 V ± 0.2 V for commercial or industrial use in FLEX 10KE devices.
(4) Operating conditions: V_{CCIO} = 2.5 V, 3.3 V, or 5.0 V.
(5) Because the RAM in the EAB is self-timed, this parameter can be ignored when the `WE` signal is registered.
(6) EAB macroparameters are internal parameters that can simplify predicting the behavior of an EAB at its boundary; these parameters are calculated by summing selected microparameters.
(7) These parameters are worst-case values for typical applications. Post-compilation timing simulation and timing analysis are required to determine actual worst-case performance.
(8) External reference timing parameters are factory-tested, worst-case values specified by Altera. A representative subset of signal paths is tested to approximate typical device applications.
(9) Contact Altera Applications for test circuit specifications and test conditions.
(10) These timing parameters are sample-tested only.
(11) Contact Altera Applications for test circuit specifications and test conditions.
(12) This parameter is measured with the measurement and test conditions, including load, specified in the ***PCI Local Bus Specification, Revision** 2.2.*

Tables 30 through 36 show EPF10K30E device internal and external timing parameters.

Table 30. EPF10K30E Device LE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{LUT}		0.6		0.8		1.1	ns
t_{CLUT}		0.5		0.6		0.8	ns
t_{RLUT}		0.7		0.8		1.1	ns
t_{PACKED}		0.5		0.6		0.8	ns
t_{EN}		0.6		0.7		0.9	ns
t_{CICO}		0.2		0.2		0.3	ns
t_{CGEN}		0.5		0.5		0.8	ns
t_{CGENR}		0.2		0.2		0.3	ns
t_{CASC}		0.8		0.9		1.2	ns
t_C		0.5		0.6		0.8	ns
t_{CO}		0.5		0.6		0.7	ns
t_{COMB}		0.5		0.6		0.7	ns
t_{SU}	0.5		0.6		0.8		ns
t_H	0.9		1.1		1.5		ns
t_{PRE}		0.5		0.6		0.8	ns
t_{CLR}		0.5		0.6		0.8	ns
t_{CH}	2.0		2.5		3.0		ns
t_{CL}	2.0		2.5		3.0		ns

Table 31. EPF10K30E Device IOE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{IOD}		2.4		2.8		3.8	ns
t_{IOC}		0.3		0.3		0.5	ns
t_{IOCO}		0.2		0.2		0.3	ns
t_{IOCOMB}		0.5		0.6		0.8	ns
t_{IOSU}	2.2		2.6		3.5		ns
t_{IOH}	0.5		0.6		0.8		ns
t_{IOCLR}		0.2		0.2		0.3	ns
t_{OD1}		1.1		1.3		1.8	ns
t_{OD2}		0.6		0.9		1.6	ns
t_{OD3}		3.0		3.5		4.8	ns
t_{XZ}		1.1		1.3		1.8	ns
t_{ZX1}		1.1		1.3		1.6	ns
t_{ZX2}		0.6		0.9		1.6	ns
t_{ZX3}		3.0		3.5		4.8	ns
t_{INREG}		5.0		5.9		8.0	ns
t_{IOFD}		3.0		3.6		4.8	ns
t_{INCOMB}		3.0		3.6		4.8	ns

Table 32. EPF10K30E Device EAB Internal Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		1.7		2.0		2.7	ns
$t_{EABDATA2}$		0.6		0.7		0.9	ns
t_{EABWE1}		1.1		1.3		1.8	ns
t_{EABWE2}		0.4		0.4		0.6	ns
t_{EABRE1}		0.8		0.9		1.2	ns
t_{EABRE2}		0.4		0.4		0.6	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		0.3		0.3		0.5	ns
$t_{EABBYPASS}$		0.5		0.6		0.8	ns
t_{EABSU}	0.9		1.0		1.4		ns
t_{EABH}	0.4		0.4		0.6		ns
t_{EABCLR}		0.3		0.3		0.5	ns
t_{AA}		3.2		3.8		5.1	ns
t_{WP}	2.5		2.9		3.9		ns
t_{RP}	0.9		1.1		1.5		ns
t_{WDSU}	0.9		1.0		1.4		ns
t_{WDH}	0.1		0.1		0.2		ns
t_{WASU}	1.7		2.0		2.7		ns
t_{WAH}	1.8		2.1		2.9		ns
t_{RASU}	3.1		3.7		5.0		ns
t_{RAH}	0.2		0.2		0.3		ns
t_{WO}		2.5		2.9		3.9	ns
t_{DD}		2.5		2.9		3.9	ns
t_{EABOUT}		0.5		0.6		0.8	ns
t_{EABCH}	1.5		2.0		2.5		ns
t_{EABCL}	1.5		2.0		2.5		ns

Table 33. EPF10K30E Device EAB Internal Timing Macroparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{EABAA}		6.4		7.6		10.2	ns
$t_{EABRCCOMB}$	6.4		7.6		10.2		ns
$t_{EABRCREG}$	4.4		5.1		7.0		ns
t_{EABWP}	2.5		2.9		3.9		ns
$t_{EABWCCOMB}$	6.0		7.0		9.5		ns
$t_{EABWCREG}$	6.8		7.8		10.6		ns
t_{EABDD}		5.7		6.7		9.0	ns
$t_{EABDATACO}$		0.8		0.9		1.3	ns
$t_{EABDATASU}$	1.5		1.7		2.3		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	1.3		1.4		2.0		ns
t_{EABWEH}	0.0		0.0		0.0		ns
$t_{EABWDSU}$	1.5		1.7		2.3		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	3.0		3.6		4.8		ns
t_{EABWAH}	0.5		0.5		0.8		ns
t_{EABWO}		5.1		6.0		8.1	ns

Table 34. EPF10K30E Device Interconnect Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		4.1		4.6		5.9	ns
t_{DIN2LE}		0.9		1.0		1.3	ns
$t_{DIN2DATA}$		1.8		1.9		2.3	ns
$t_{DCLK2IOE}$		3.9		4.6		6.2	ns
$t_{DCLK2LE}$		0.9		1.0		1.3	ns
$t_{SAMELAB}$		0.1		0.1		0.2	ns
$t_{SAMEROW}$		1.3		1.3		1.8	ns
$t_{SAMECOLUMN}$		0.7		0.8		1.5	ns
$t_{DIFFROW}$		2.0		2.1		3.3	ns
$t_{TWOROWS}$		3.3		3.4		5.1	ns
$t_{LEPERIPH}$		3.8		4.1		5.3	ns
$t_{LABCARRY}$		0.1		0.1		0.2	ns
$t_{LABCASC}$		0.3		0.3		0.5	ns

Table 35. EPF10K30E External Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{DRR}		8.5		10.0		13.5	ns
t_{INSU} *(3)*	3.0		3.6		4.8		ns
t_{INH} *(3)*	0.0		0.0		0.0		ns
t_{OUTCO} *(3)*	2.0	3.5	2.0	4.5	2.0	7.1	ns
t_{INSU} *(4)*	2.0		2.6		–		ns
t_{INH} *(4)*	0.0		0.0		–	–	ns
t_{OUTCO} *(4)*	0.5	2.5	0.5	3.5	–	–	ns
t_{PCISU}	3.0		4.2		6.4		ns
t_{PCIH}	0.0		0.0		0.0		ns
t_{PCICO}	2.0	6.0	2.0	7.5	2.0	10.2	ns

Table 36. EPF10K30E External Bidirectional Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	1.5		2.2		3.6		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$ *(3)*	2.0	3.5	2.0	4.5	2.0	7.1	ns
$t_{XZBIDIR}$ *(3)*	2.0	5.8	2.0	6.3	2.0	8.0	ns
$t_{ZXBIDIR}$ *(3)*	2.0	4.7	2.0	5.3	2.0	7.2	ns
$t_{OUTCOBIDIR}$ *(4)*	0.5	2.5	0.5	3.5	–	–	ns
$t_{XZBIDIR}$ *(4)*	0.5	5.3	0.5	5.8	–	–	ns
$t_{ZXBIDIR}$ *(4)*	0.5	4.2	0.5	4.8	–	–	ns

Notes to tables:

(1) All timing parameters are described in Tables 22 through 29 in this data sheet.
(2) These parameters are specified by characterization.
(3) This parameter is measured without the use of the ClockLock or ClockBoost circuits.
(4) This parameter is measured with the use of the ClockLock or ClockBoost circuits.

Tables 37 through 43 show EPF10K50E device internal and external timing parameters.

Table 37. EPF10K50E Device LE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{LUT}		0.6		0.8		1.1	ns
t_{CLUT}		0.5		0.6		0.8	ns
t_{RLUT}		0.7		0.8		1.1	ns
t_{PACKED}		0.5		0.6		0.8	ns
t_{EN}		0.6		0.7		0.9	ns
t_{CICO}		0.2		0.2		0.3	ns
t_{CGEN}		0.5		0.5		0.8	ns
t_{CGENR}		0.2		0.2		0.3	ns
t_{CASC}		0.8		0.9		1.2	ns
t_C		0.5		0.6		0.8	ns
t_{CO}		0.5		0.6		0.7	ns
t_{COMB}		0.5		0.6		0.7	ns
t_{SU}	0.5		0.6		0.8		ns
t_H	0.9		1.1		1.5		ns
t_{PRE}		0.5		0.6		0.8	ns
t_{CLR}		0.5		0.6		0.8	ns
t_{CH}	2.0		2.5		3.0		ns
t_{CL}	2.0		2.5		3.0		ns

Table 38. EPF10K50E Device IOE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{IOD}		2.4		2.8		3.8	ns
t_{IOC}		0.3		0.3		0.5	ns
t_{IOCO}		0.2		0.2		0.3	ns
t_{IOCOMB}		0.5		0.6		0.8	ns
t_{IOSU}	2.2		2.6		3.5		ns
t_{IOH}	0.5		0.6		0.8		ns
t_{IOCLR}		0.2		0.2		0.3	ns
t_{OD1}		1.1		1.3		1.8	ns
t_{OD2}		0.6		0.9		1.6	ns
t_{OD3}		3.0		3.5		4.8	ns
t_{XZ}		1.1		1.3		1.8	ns
t_{ZX1}		1.1		1.3		1.6	ns
t_{ZX2}		0.6		0.9		1.6	ns
t_{ZX3}		3.0		3.5		4.8	ns
t_{INREG}		5.0		5.9		8.0	ns
t_{IOFD}		3.0		3.6		4.8	ns
t_{INCOMB}		3.0		3.6		4.8	ns

Table 39. EPF10K50E Device EAB Internal Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		1.7		2.0		2.7	ns
$t_{EABDATA2}$		0.6		0.7		0.9	ns
t_{EABWE1}		1.1		1.3		1.8	ns
t_{EABWE2}		0.4		0.4		0.6	ns
t_{EABRE1}		0.8		0.9		1.2	ns
t_{EABRE2}		0.4		0.4		0.6	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		0.3		0.3		0.5	ns
$t_{EABBYPASS}$		0.5		0.6		0.8	ns
t_{EABSU}	0.9		1.0		1.4		ns
t_{EABH}	0.4		0.4		0.6		ns
t_{EABCLR}		0.3		0.3		0.5	ns
t_{AA}		3.2		3.8		5.1	ns
t_{WP}	2.5		2.9		3.9		ns
t_{RP}	0.9		1.1		1.5		ns
t_{WDSU}	0.9		1.0		1.4		ns
t_{WDH}	0.1		0.1		0.2		ns
t_{WASU}	1.7		2.0		2.7		ns
t_{WAH}	1.8		2.1		2.9		ns
t_{RASU}	3.1		3.7		5.0		ns
t_{RAH}	0.2		0.2		0.3		ns
t_{WO}		2.5		2.9		3.9	ns
t_{DD}		2.5		2.9		3.9	ns
t_{EABOUT}		0.5		0.6		0.8	ns
t_{EABCH}	1.5		2.0		2.5		ns
t_{EABCL}	1.5		2.0		2.5		ns

Table 40. EPF10K50E Device EAβ Internal Timing Macroparameters *Note (1)*

Symbol	**Speed Grade**						**Unit**
	-1		**-2**		**-3**		
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{EABAA}		6.4		7.6		10.2	ns
$t_{EABRCCOMB}$	6.4		7.6		10.2		ns
$t_{EABRCREG}$	4.4		5.1		7.0		ns
t_{EABWP}	2.5		2.9		3.9		ns
$t_{EABWCCOMB}$	6.0		7.0		9.5		ns
$t_{EABWCREG}$	6.8		7.8		10.6		ns
t_{EABDD}		5.7		6.7		9.0	ns
$t_{EABDATACO}$		0.8		0.9		1.3	ns
$t_{EABDATASU}$	1.5		1.7		2.3		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	1.3		1.4		2.0		ns
t_{EABWEH}	0.0		0.0		0.0		ns
$t_{EABWDSU}$	1.5		1.7		2.3		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	3.0		3.6		4.8		ns
t_{EABWAH}	0.5		0.5		0.8		ns
t_{EABWO}		5.1		6.0		8.1	ns

Table 41. EPF10K50E Device Interconnect Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		4.6		5.1		6.4	ns
t_{DIN2LE}		0.9		1.0		1.3	ns
$t_{DIN2DATA}$		1.5		1.9		2.3	ns
$t_{DCLK2IOE}$		3.9		4.6		6.2	ns
$t_{DCLK2LE}$		0.9		1.0		1.3	ns
$t_{SAMELAB}$		0.1		0.1		0.2	ns
$t_{SAMEROW}$		1.3		1.3		1.8	ns
$t_{SAMECOLUMN}$		1.6		1.7		2.4	ns
$t_{DIFFROW}$		2.9		3.0		4.2	ns
$t_{TWOROWS}$		4.2		4.3		6.0	ns
$t_{LEPERIPH}$		4.3		4.6		5.8	ns
$t_{LABCARRY}$		0.1		0.1		0.2	ns
$t_{LABCASC}$		0.3		0.3		0.5	ns

Table 42. EPF10K50E External Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{DRR}		8.5		10.0		13.5	ns
t_{INSU}	3.0		3.6		4.8		ns
t_{INH}	0.0		0.0		0.0		ns
t_{OUTCO}	2.0	4.6	2.0	5.4	2.0	8.0	ns
t_{PCISU}	3.0		4.2		6.4		ns
t_{PCIH}	0.0		0.0		0.0		ns
t_{PCICO}	2.0	6.0	2.0	7.7	2.0	10.5	ns

Table 43. EPF10K50E External βidirectional Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	2.7		3.4		4.8		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$	2.0	4.6	2.0	5.4	2.0	8.2	ns
$t_{XZBIDIR}$	2.0	6.3	2.0	6.8	2.0	8.5	ns
$t_{ZXBIDIR}$	2.0	5.2	2.0	5.8	2.0	7.7	ns

Notes to tables:

(1) All timing parameters are described in Tables 22 through 29 in this data sheet.
(2) These parameters are specified by characterization.

Tables 44 through 50 show EPF10K100E device internal and external timing parameters.

Table 44. EPF10K100E Device LE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{LUT}		0.6		0.8		1.1	ns
t_{CLUT}		0.5		0.6		0.8	ns
t_{RLUT}		0.7		0.8		1.1	ns
t_{PACKED}		0.5		0.6		0.8	ns
t_{EN}		0.6		0.7		0.9	ns
t_{CICO}		0.2		0.2		0.3	ns
t_{CGEN}		0.5		0.5		0.8	ns
t_{CGENR}		0.2		0.2		0.3	ns
t_{CASC}		0.8		0.9		1.2	ns
t_{C}		0.5		0.6		0.8	ns
t_{CO}		0.5		0.6		0.7	ns
t_{COMB}		0.5		0.6		0.7	ns
t_{SU}	0.5		0.6		0.8		ns
t_{H}	0.9		1.1		1.5		ns
t_{PRE}		0.5		0.6		0.8	ns
t_{CLR}		0.5		0.6		0.8	ns
t_{CH}	2.0		2.5		3.0		ns
t_{CL}	2.0		2.5		3.0		ns

Table 45. EPF10K100E Device IOE Timing Microparameters Note (1)							
Symbol	**Speed Grade**						**Unit**
	-1		**-2**		**-3**		
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{IOD}		2.4		2.8		3.8	ns
t_{IOC}		0.3		0.3		0.5	ns
t_{IOCO}		0.2		0.2		0.3	ns
t_{IOCOMB}		0.5		0.6		0.8	ns
t_{IOSU}	2.2		2.6		3.5		ns
t_{IOH}	0.5		0.6		0.8		ns
t_{IOCLR}		0.2		0.2		0.3	ns
t_{OD1}		1.1		1.3		1.8	ns
t_{OD2}		0.6		0.9		1.6	ns
t_{OD3}		3.0		3.5		4.8	ns
t_{XZ}		1.1		1.3		1.8	ns
t_{ZX1}		1.1		1.3		1.6	ns
t_{ZX2}		0.6		0.9		1.6	ns
t_{ZX3}		3.0		3.5		4.8	ns
t_{INREG}		5.0		5.9		8.0	ns
t_{IOFD}		3.0		3.6		4.8	ns
t_{INCOMB}		3.0		3.6		4.8	ns

Table 46. EPF10K100E Device EAB Internal Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		1.7		2.0		2.7	ns
$t_{EABDATA2}$		0.6		0.7		0.9	ns
t_{EABWE1}		1.1		1.3		1.8	ns
t_{EABWE2}		0.4		0.4		0.6	ns
t_{EABRE1}		0.8		0.9		1.2	ns
t_{EABRE2}		0.4		0.4		0.6	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		0.3		0.3		0.5	ns
$t_{EABBYPASS}$		0.5		0.6		0.8	ns
t_{EABSU}	0.9		1.0		1.4		ns
t_{EABH}	0.4		0.4		0.6		ns
t_{EABCLR}		0.3		0.3		0.5	ns
t_{AA}		3.2		3.8		5.1	ns
t_{WP}	2.5		2.9		3.9		ns
t_{RP}	0.9		1.1		1.5		ns
t_{WDSU}	0.9		1.0		1.4		ns
t_{WDH}	0.1		0.1		0.2		ns
t_{WASU}	1.7		2.0		2.7		ns
t_{WAH}	1.8		2.1		2.9		ns
t_{RASU}	3.1		3.7		5.0		ns
t_{RAH}	0.2		0.2		0.3		ns
t_{WO}		2.5		2.9		3.9	ns
t_{DD}		2.5		2.9		3.9	ns
t_{EABOUT}		0.5		0.6		0.8	ns
t_{EABCH}	1.5		2.0		2.5		ns
t_{EABCL}	1.5		2.0		2.5		ns

Table 47. EPF10K100E Device EAB Internal Timing Macroparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{EABAA}		6.4		7.6		10.2	ns
$t_{EABRCCOMB}$	6.4		7.6		10.2		ns
$t_{EABRCREG}$	4.4		5.1		7.0		ns
t_{EABWP}	2.5		2.9		3.9		ns
$t_{EABWCCOMB}$	6.0		7.0		9.5		ns
$t_{EABWCREG}$	6.8		7.8		10.6		ns
t_{EABDD}		5.7		6.7		9.0	ns
$t_{EABDATACO}$		0.8		0.9		1.3	ns
$t_{EABDATASU}$	1.5		1.7		2.3		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	1.3		1.4		2.0		ns
t_{EABWEH}	0.0		0.0		0.0		ns
$t_{EABWDSU}$	1.5		1.7		2.3		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	3.0		3.6		4.8		ns
t_{EABWAH}	0.5		0.5		0.8		ns
t_{EABWO}		5.1		6.0		8.1	ns

Table 48. EPF10K100E Device Interconnect Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		3.5		3.9		4.9	ns
t_{DIN2LE}		0.6		0.6		0.9	ns
$t_{DIN2DATA}$		2.0		2.1		2.9	ns
$t_{DCLK2IOE}$		1.7		2.0		2.8	ns
$t_{DCLK2LE}$		0.6		0.6		0.9	ns
$t_{SAMELAB}$		0.1		0.1		0.2	ns
$t_{SAMEROW}$		1.7		1.8		1.7	ns
$t_{SAMECOLUMN}$		1.2		1.1		0.8	ns
$t_{DIFFROW}$		2.9		2.9		2.5	ns
$t_{TWOROWS}$		4.6		4.7		4.2	ns
$t_{LEPERIPH}$		4.3		4.9		5.9	ns
$t_{LABCARRY}$		0.1		0.1		0.2	ns
$t_{LABCASC}$		0.3		0.3		0.5	ns

Table 49. EPF10K100E External Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{DRR}		10.0		12.0		16.0	ns
t_{INSU} *(3)*	3.1		3.7		4.7		ns
t_{INH} *(3)*	0.0		0.0		0.0		ns
t_{OUTCO} *(3)*	2.0	3.7	2.0	4.4	2.0	6.3	ns
t_{INSU} *(4)*	2.1		2.7		–		ns
t_{INH} *(4)*	0.0		0.0		–		ns
t_{OUTCO} *(4)*	2.0	2.7	0.5	3.4	–	–	ns
t_{PCISU}	3.0		6.2		–		ns
t_{PCIH}	0.0		0.0		–		ns
t_{PCICO}	2.0	6.0		6.9	–	–	ns

Table 50. EPF10K100E External Bidirectional Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	2.5		3.3		4.4		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$ *(3)*	2.0	3.7	2.0	4.4	2.0	6.3	ns
$t_{XZBIDIR}$ *(3)*	2.0	5.2	2.0	6.1	2.0	8.3	ns
$t_{ZXBIDIR}$ *(3)*	2.0	4.7	2.0	5.6	2.0	8.1	ns
$t_{OUTCOBIDIR}$ *(4)*	0.5	2.7	0.5	3.4	–	–	ns
$t_{XZBIDIR}$ *(4)*	0.5	4.2	0.5	5.1	–	–	ns
$t_{ZXBIDIR}$ *(4)*	0.5	3.7	0.5	4.6	–	–	ns

Notes to tables:

(1) All timing parameters are described in Tables 22 through 29 in this data sheet.
(2) These parameters are specified by characterization.
(3) This parameter is measured without the use of the ClockLock or ClockBoost circuits.
(4) This parameter is measured with the use of the ClockLock or ClockBoost circuits.

Tables 51 through 57 show EPF10K130E device internal and external timing parameters.

Table 51. EPF10K130E Device LE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{LUT}		0.6		0.8		1.1	ns
t_{CLUT}		0.5		0.6		0.8	ns
t_{RLUT}		0.7		0.8		1.1	ns
t_{PACKED}		0.5		0.6		0.8	ns
t_{EN}		0.6		0.7		0.9	ns
t_{CICO}		0.2		0.2		0.3	ns
t_{CGEN}		0.5		0.5		0.8	ns
t_{CGENR}		0.2		0.2		0.3	ns
t_{CASC}		0.8		0.9		1.2	ns
t_C		0.5		0.6		0.8	ns
t_{CO}		0.5		0.6		0.7	ns
t_{COMB}		0.5		0.6		0.7	ns
t_{SU}	0.5		0.6		0.8		ns
t_H	0.9		1.1		1.5		ns
t_{PRE}		0.5		0.6		0.8	ns
t_{CLR}		0.5		0.6		0.8	ns
t_{CH}	2.0		2.5		3.0		ns
t_{CL}	2.0		2.5		3.0		ns

Table 52. EPF10K130E Device IOE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{IOD}		2.4		2.8		3.8	ns
t_{IOC}		0.3		0.3		0.5	ns
t_{IOCO}		0.2		0.2		0.3	ns
t_{IOCOMB}		0.5		0.6		0.8	ns
t_{IOSU}	2.2		2.6		3.5		ns
t_{IOH}	0.5		0.6		0.8		ns
t_{IOCLR}		0.2		0.2		0.3	ns
t_{OD1}		1.1		1.3		1.8	ns
t_{OD2}		0.6		0.9		1.6	ns
t_{OD3}		3.0		3.5		4.8	ns
t_{XZ}		1.1		1.3		1.8	ns
t_{ZX1}		1.1		1.3		1.6	ns
t_{ZX2}		0.6		0.9		1.6	ns
t_{ZX3}		3.0		3.5		4.8	ns
t_{INREG}		5.0		5.9		8.0	ns
t_{IOFD}		3.0		3.6		4.8	ns
t_{INCOMB}		3.0		3.6		4.8	ns

Table 53. EPF10K130E Device EAB Internal Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		1.7		2.0		2.7	ns
$t_{EABDATA2}$		0.6		0.7		0.9	ns
t_{EABWE1}		1.1		1.3		1.8	ns
t_{EABWE2}		0.4		0.4		0.6	ns
t_{EABRE1}		0.8		0.9		1.2	ns
t_{EABRE2}		0.4		0.4		0.6	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		0.3		0.3		0.5	ns
$t_{EABBYPASS}$		0.5		0.6		0.8	ns
t_{EABSU}	0.9		1.0		1.4		ns
t_{EABH}	0.4		0.4		0.6		ns
t_{EABCLR}		0.3		0.3		0.5	ns
t_{AA}		3.2		3.8		5.1	ns
t_{WP}	2.5		2.9		3.9		ns
t_{RP}	0.9		1.1		1.5		ns
t_{WDSU}	0.9		1.0		1.4		ns
t_{WDH}	0.1		0.1		0.2		ns
t_{WASU}	1.7		2.0		2.7		ns
t_{WAH}	1.8		2.1		2.9		ns
t_{RASU}	3.1		3.7		5.0		ns
t_{RAH}	0.2		0.2		0.3		ns
t_{WO}		2.5		2.9		3.9	ns
t_{DD}		2.5		2.9		3.9	ns
t_{EABOUT}		0.5		0.6		0.8	ns
t_{EABCH}	1.5		2.0		2.5		ns
t_{EABCL}	1.5		2.0		2.5		ns

Table 54. EPF10K130E Device EAβ Internal Timing Macroparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{EABAA}		6.4		7.6		10.2	ns
$t_{EABRCCOMB}$	6.4		7.6		10.2		ns
$t_{EABRCREG}$	4.4		5.1		7.0		ns
t_{EABWP}	2.5		2.9		3.9		ns
$t_{EABWCCOMB}$	6.0		7.0		9.5		ns
$t_{EABWCREG}$	6.8		7.8		10.6		ns
t_{EABDD}		5.7		6.7		9.0	ns
$t_{EABDATACO}$		0.8		0.9		1.3	ns
$t_{EABDATASU}$	1.5		1.7		2.3		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	1.3		1.4		2.0		ns
t_{EABWEH}	0.0		0.0		0.0		ns
$t_{EABWDSU}$	1.5		1.7		2.3		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	3.0		3.6		4.8		ns
t_{EABWAH}	0.5		0.5		0.8		ns
t_{EABWO}		5.1		6.0		8.1	ns

Table 55. EPF10K130E Device Interconnect Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		3.9		4.3		5.3	ns
t_{DIN2LE}		0.6		0.6		0.9	ns
$t_{DIN2DATA}$		2.0		2.1		2.9	ns
$t_{DCLK2IOE}$		1.7		2.0		2.8	ns
$t_{DCLK2LE}$		0.6		0.6		0.9	ns
$t_{SAMELAB}$		0.1		0.1		0.2	ns
$t_{SAMEROW}$		1.7		1.8		1.7	ns
$t_{SAMECOLUMN}$		1.6		1.5		1.4	ns
$t_{DIFFROW}$		3.3		3.3		3.1	ns
$t_{TWOROWS}$		5.0		5.1		4.8	ns
$t_{LEPERIPH}$		4.7		5.3		6.3	ns
$t_{LABCARRY}$		0.1		0.1		0.2	ns
$t_{LABCASC}$		0.3		0.3		0.5	ns

Table 56. EPF10K130E External Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{DRR}		10.0		12.0		16.0	ns
t_{INSU} *(3)*	3.1		3.7		4.7		ns
t_{INH} *(3)*	0.0		0.0		0.0		ns
t_{OUTCO} *(3)*	2.0	3.7	2.0	4.4	2.0	6.3	ns
t_{INSU} *(4)*	2.1		2.7		–		ns
t_{INH} *(4)*	0.0		0.0		–		ns
t_{OUTCO} *(4)*	2.0	2.7	0.5	3.4	–	–	ns
t_{PCISU}	3.0		6.2		–		ns
t_{PCIH}	0.0		0.0		–		ns
t_{PCICO}	2.0	6.0	2.0	6.9	–	–	ns

Table 57. EPF10K130E External Bidirectional Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	2.5		3.3		4.4		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$ (3)	2.0	3.7	2.0	4.4	2.0	6.3	ns
$t_{XZBIDIR}$ (3)		4.2		4.9		6.8	ns
$t_{ZXBIDIR}$ (3)		3.7		4.5		6.6	ns
$t_{OUTCOBIDIR}$ (4)	0.5	2.7	0.5	3.4	–	–	ns
$t_{XZBIDIR}$ (4)	0.5	4.2	0.5	5.1	–	–	ns
$t_{ZXBIDIR}$ (4)	0.5	3.7	0.5	4.6	–	–	ns

Notes to tables:

(1) All timing parameters are described in Tables 22 through 29 in this data sheet.
(2) These parameters are specified by characterization.
(3) This parameter is measured without the use of the ClockLock or ClockBoost circuits.
(4) This parameter is measured with the use of the ClockLock or ClockBoost circuits.

Tables 58 through 64 show EPF10K200E device internal and external timing parameters.

Table 58. EPF10K200E Device LE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{LUT}		0.7		0.8		1.2	ns
t_{CLUT}		0.4		0.5		0.6	ns
t_{RLUT}		0.6		0.7		0.9	ns
t_{PACKED}		0.3		0.3		0.5	ns
t_{EN}		0.4		0.5		0.6	ns
t_{CICO}		0.2		0.2		0.3	ns
t_{CGEN}		0.4		0.4		0.6	ns
t_{CGENR}		0.2		0.2		0.3	ns
t_{CASC}		0.7		0.8		1.2	ns
t_C		0.5		0.6		0.8	ns
t_{CO}		0.5		0.6		0.8	ns
t_{COMB}		0.4		0.6		0.8	ns
t_{SU}	0.4		0.6		0.7		ns
t_H	0.9		1.1		1.5		ns
t_{PRE}		0.5		0.6		0.8	ns
t_{CLR}		0.5		0.6		0.8	ns
t_{CH}	2.0		2.5		3.0		ns
t_{CL}	2.0		2.5		3.0		ns

Table 59. EPF10K200E Device IOE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{IOD}		2.7		3.3		4.4	ns
t_{IOC}		0.3		0.3		0.5	ns
t_{IOCO}		1.6		1.9		2.6	ns
t_{IOCOMB}		0.5		0.6		0.8	ns
t_{IOSU}	0.8		0.9		1.2		ns
t_{IOH}	0.7		0.8		1.1		ns
t_{IOCLR}		0.2		0.2		0.3	ns
t_{OD1}		0.6		0.7		0.9	ns
t_{OD2}		0.1		0.2		0.7	ns
t_{OD3}		2.5		3.0		3.9	ns
t_{XZ}		2.1		2.5		3.3	ns
t_{ZX1}		2.1		2.5		3.3	ns
t_{ZX2}		1.6		2.0		3.1	ns
t_{ZX3}		4.0		4.8		6.3	ns
t_{INREG}		4.3		5.1		6.8	ns
t_{IOFD}		0.5		0.6		0.8	ns
t_{INCOMB}		0.5		0.6		0.8	ns

Table 60. EPF10K200E Device EAB Internal Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		2.0		2.4		3.2	ns
$t_{EABDATA2}$		0.4		0.5		0.6	ns
t_{EABWE1}		1.4		1.7		2.3	ns
t_{EABWE2}		0.0		0.0		0.0	ns
t_{EABRE1}		0.0		0.0		0.0	ns
t_{EABRE2}		0.4		0.5		0.6	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		0.8		0.9		1.2	ns
$t_{EABBYPASS}$		0.0		0.1		0.1	ns
t_{EABSU}	0.9		1.1		1.5		ns
t_{EABH}	0.4		0.5		0.6		ns
t_{EABCLR}		0.8		0.9		1.2	ns
t_{AA}		3.1		3.7		4.9	ns
t_{WP}	3.3		4.0		5.3		ns
t_{RP}	0.9		1.1		1.5		ns
t_{WDSU}	0.9		1.1		1.5		ns
t_{WDH}	0.1		0.1		0.1		ns
t_{WASU}	1.3		1.6		2.1		ns
t_{WAH}	2.1		2.5		3.3		ns
t_{RASU}	2.2		2.6		3.5		ns
t_{RAH}	0.1		0.1		0.2		ns
t_{WO}		2.0		2.4		3.2	ns
t_{DD}		2.0		2.4		3.2	ns
t_{EABOUT}		0.0		0.1		0.1	ns
t_{EABCH}	1.5		2.0		2.5		ns
t_{EABCL}	1.5		2.0		2.5		ns

Table 61. EPF10K200E Device EAβ Internal Timing Macroparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{EABAA}		5.1		6.4		8.4	ns
$t_{EABRCCOMB}$	5.1		6.4		8.4		ns
$t_{EABRCREG}$	4.8		5.7		7.6		ns
t_{EABWP}	3.3		4.0		5.3		ns
$t_{EABWCCOMB}$	6.7		8.1		10.7		ns
$t_{EABWCREG}$	6.6		8.0		10.6		ns
t_{EABDD}		4.0		5.1		6.7	ns
$t_{EABDATACO}$		0.8		1.0		1.3	ns
$t_{EABDATASU}$	1.3		1.6		2.1		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	0.9		1.1		1.5		ns
t_{EABWEH}	0.4		0.5		0.6		ns
$t_{EABWDSU}$	1.5		1.8		2.4		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	3.0		3.6		4.7		ns
t_{EABWAH}	0.4		0.5		0.7		ns
t_{EABWO}		3.4		4.4		5.8	ns

Table 62. EPF10K200E Device Interconnect Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		4.9		5.3		6.3	ns
t_{DIN2LE}		0.6		0.6		0.9	ns
$t_{DIN2DATA}$		1.9		2.1		2.8	ns
$t_{DCLK2IOE}$		1.7		2.0		2.8	ns
$t_{DCLK2LE}$		0.6		0.6		0.9	ns
$t_{SAMELAB}$		0.1		0.1		0.2	ns
$t_{SAMEROW}$		2.9		3.2		3.9	ns
$t_{SAMECOLUMN}$		3.3		3.5		4.6	ns
$t_{DIFFROW}$		6.2		6.7		8.5	ns
$t_{TWOROWS}$		9.1		9.9		12.4	ns
$t_{LEPERIPH}$		5.6		6.2		7.2	ns
$t_{LABCARRY}$		0.1		0.1		0.2	ns
$t_{LABCASC}$		0.9		1.0		1.4	ns

Table 63. EPF10K200E External Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{DRR}		10.0		12.0		16.0	ns
t_{INSU} *(3)*	3.1		3.7		4.7		ns
t_{INH} *(3)*	0.0		0.0		0.0		ns
t_{OUTCO} *(3)*	2.0	3.7	2.0	4.4	2.0	6.3	ns
t_{INSU} *(4)*	2.1		2.7		–		ns
t_{INH} *(4)*	0.0		0.0		–		ns
t_{OUTCO} *(4)*	0.5	3.0	0.5	3.4	–	–	ns
t_{PCISU}	3.0		6.2		–		ns
t_{PCIH}	0.0		0.0		–		ns
t_{PCICO}	2.0	6.0	2.0	8.9	–	–	ns

Table 64. EPF10K200E External Bidirectional Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{INSUBDIR}$ *(3)*	2.5		3.3		4.4		ns
$t_{INHBIDIR}$ *(3)*	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$ *(3)*	2.0	3.7	2.0	4.4	2.0	6.3	ns
$t_{XZBIDIR}$ *(3)*	2.0	5.2	2.0	6.1	2.0	8.3	ns
$t_{ZXBIDIR}$ *(3)*	2.0	4.7	2.0	5.6	2.0	8.1	ns
$t_{INSUBDIR}$ *(4)*	3.5		4.3		–		ns
$t_{INHBIDIR}$ *(4)*	0.0		0.0		–		ns
$t_{OUTCOBIDIR}$ *(4)*	0.5	2.7	0.5	3.4	–	–	ns
$t_{XZBIDIR}$ *(4)*	0.5	4.2	0.5	5.1		–	ns
$t_{ZXBIDIR}$ *(4)*	0.5	3.7	0.5	4.6		–	ns

Notes to tables:

(1) All timing parameters are described in Tables 22 through 29 in this data sheet.
(2) These parameters are specified by characterization.
(3) This parameter is measured without the use of the ClockLock or ClockBoost circuits.
(4) This parameter is measured with the use of the ClockLock or ClockBoost circuits.

Tables 65 through 71 show EPF10K100B device internal and external timing parameters.

Table 65. EPF10K100β Device LE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{LUT}		1.7		1.9		2.1	ns
t_{CLUT}		0.8		0.9		0.9	ns
t_{RLUT}		1.3		1.4		1.6	ns
t_{PACKED}		0.6		0.7		0.8	ns
t_{EN}		0.5		0.5		0.7	ns
t_{CICO}		0.2		0.2		0.3	ns
t_{CGEN}		0.5		0.5		0.7	ns
t_{CGENR}		0.5		0.5		0.7	ns
t_{CASC}		0.6		0.6		0.8	ns
t_{C}		1.0		1.1		1.1	ns
t_{CO}		0.6		0.6		0.8	ns
t_{COMB}		0.5		0.6		0.8	ns
t_{SU}	0.5		0.6		0.8		ns
t_{H}	1.5		1.6		2.0		ns
t_{PRE}		0.6		0.7		0.8	ns
t_{CLR}		0.6		0.7		0.8	ns
t_{CH}	2.0		2.5		3.0		ns
t_{CL}	2.0		2.5		3.0		ns

Table 66. EPF10K100β Device IOE Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{IOD}		1.5		1.6		1.7	ns
t_{IOC}		0.3		0.3		0.4	ns
t_{IOCO}		0.2		0.2		0.3	ns
t_{IOCOMB}		0.5		0.5		0.7	ns
t_{IOSU}	1.2		1.3		1.6		ns
t_{IOH}	1.2		1.3		1.6		ns
t_{IOCLR}		0.0		0.0		0.0	ns
t_{OD1}		2.3		2.5		2.5	ns
t_{OD2}		3.1		3.4		3.6	ns
t_{OD3}		4.3		4.7		5.2	ns
t_{XZ}		4.5		4.9		5.9	ns
t_{ZX1}		4.5		4.9		5.9	ns
t_{ZX2}		3.1		3.9		5.4	ns
t_{ZX3}		6.5		7.1		8.6	ns
t_{INREG}		6.0		6.5		7.9	ns
t_{IOFD}		2.8		3.1		3.7	ns
t_{INCOMB}		2.8		3.1		4.8	ns

Table 67. EPF10K100β Device EAβ Internal Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{EABDATA1}$		3.2		3.5		4.2	ns
$t_{EABDATA2}$		0.6		0.7		0.7	ns
t_{EABWE1}		1.3		1.4		1.7	ns
t_{EABWE2}		1.1		1.2		1.5	ns
t_{EABCLK}		0.0		0.0		0.0	ns
t_{EABCO}		1.0		1.1		1.3	ns
$t_{EABBYPASS}$		0.0		0.0		0.0	ns
t_{EABSU}	1.5		1.6		2.0		ns
t_{EABH}	0.1		0.1		0.1		ns
t_{AA}		3.2		3.8		5.1	ns
t_{WP}	3.4		3.7		4.5		ns
t_{WDSU}	0.1		0.1		0.1		ns
t_{WDH}	1.5		1.6		1.7		ns
t_{WASU}	1.6		1.7		2.1		ns
t_{WAH}	0.9		1.0		1.2		ns
t_{WO}		2.8		3.1		3.6	ns
t_{DD}		2.8		3.1		3.6	ns
t_{EABOUT}		0.5		0.5		0.7	ns
t_{EABCH}	2.0		2.5		3.0		ns
t_{EABCL}	3.4		3.7		4.5		ns

Table 68. EPF10K100β Device EAβ Internal Timing Macroparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{EABAA}		7.5		8.1		9.9	ns
$t_{EABRCCOMB}$	7.5		8.1		9.9		ns
$t_{EABRCREG}$	5.3		5.7		7.0		ns
t_{EABWP}	3.4		3.7		4.5		ns
$t_{EABWCCOMB}$	5.9		6.4		7.8		ns
$t_{EABWCREG}$	8.6		9.4		11.2		ns
t_{EABDD}		6.5		7.1		8.5	ns
$t_{EABDATACO}$		1.5		1.6		2.0	ns
$t_{EABDATASU}$	2.1		2.3		2.7		ns
$t_{EABDATAH}$	0.0		0.0		0.0		ns
$t_{EABWESU}$	2.6		2.8		3.5		ns
t_{EABWEH}	0.0		0.0		0.0		ns
$t_{EABWDSU}$	2.0		2.2		2.6		ns
t_{EABWDH}	0.0		0.0		0.0		ns
$t_{EABWASU}$	4.5		4.8		6.0		ns
t_{EABWAH}	0.0		0.0		0.0		ns
t_{EABWO}		4.6		5.0		6.0	ns

Table 69. EPF10K100β Device Interconnect Timing Microparameters *Note (1)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{DIN2IOE}$		4.8		4.9		5.5	ns
t_{DIN2LE}		1.1		1.2		1.7	ns
$t_{DIN2DATA}$		1.5		1.4		1.9	ns
$t_{DCLK2IOE}$		3.4		3.7		4.5	ns
$t_{DCLK2LE}$		1.1		1.2		1.7	ns
$t_{SAMELAB}$		0.3		0.3		0.4	ns
$t_{SAMEROW}$		2.0		2.4		3.3	ns
$t_{SAMECOLUMN}$		1.7		2.1		2.9	ns
$t_{DIFFROW}$		3.7		4.5		6.2	ns
$t_{TWOROWS}$		5.7		6.9		9.5	ns
$t_{LEPERIPH}$		4.8		4.8		5.2	ns
$t_{LABCARRY}$		0.0		0.0		0.0	ns
$t_{LABCASC}$		0.4		0.4		0.5	ns

Table 70. EPF10K100β External Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{DRR}		11.0		12.0		14.5	ns
t_{INSU}	3.5		3.8		4.6		ns
t_{INH}	0.0		0.0		0.0		ns
t_{OUTCO}	2.0	4.8	2.0	5.7	2.0	7.2	ns

Table 71. EPF10K100β External βidirectional Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{INSUBIDIR}$	4.1		4.5		5.5		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$	2.0	4.8	2.0	5.7	2.0	7.2	ns
$t_{XZBIDIR}$		8.4		9.0		11.0	ns
$t_{ZXBIDIR}$		8.4		9.0		11.0	ns

Notes to tables:

(1) All timing parameters are described in Tables 22 through 29 in this data sheet.

(2) These parameters are specified by characterization.

Tables 72 through 75 show EPF10K50S and EPF10K200S device external timing parameters.

Table 72. EPF10K50S External Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
t_{DRR}		8.5		10.0		13.5	ns
t_{INSU} *(3)*	3.0		3.6		4.8		ns
t_{INH} *(3)*	0.0		0.0		0.0		ns
t_{OUTCO} *(3)*	2.0	3.5	2.0	4.5	2.0	7.1	ns
t_{INSU} *(4)*	2.0		2.6		–		ns
t_{INH} *(4)*	0.0		0.0		–		ns
t_{OUTCO} *(4)*	0.5	2.5	0.5	3.5	–	–	ns
t_{PCISU}	3.0		4.2		6.4		ns
t_{PCIH}	0.0		0.0		0.0		ns
t_{PCICO}	2.0	6.0	2.0	7.7	2.0	10.5	ns

Table 73. EPF10K50S External Bidirectional Timing Parameters *Notes (1), (2)*							
Symbol	**Speed Grade**						**Unit**
	-1		**-2**		**-3**		
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	
$t_{INSUBDIR}$	1.5		2.2		3.6		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$ *(3)*	2.0	3.5	2.0	4.5	2.0	7.1	ns
$t_{XZBIDIR}$ *(3)*	2.0	5.8	2.0	6.3	2.0	8.0	ns
$t_{ZXBIDIR}$ *(3)*	2.0	4.7	2.0	5.3	2.0	7.2	ns
$t_{OUTCOBIDIR}$ *(4)*	0.5	2.5	0.5	3.5	–	–	ns
$t_{XZBIDIR}$ *(4)*	0.5	4.8	0.5	5.3	–	–	ns
$t_{ZXBIDIR}$ *(4)*	0.5	3.7	0.5	4.3	–	–	ns

Table 74. EPF10K200S External Timing Parameters *Notes (1), (2)*							
Symbol	**Speed Grade**						**Unit**
	-1		**-2**		**-3**		
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{DRR}		10.0		12.0		16.0	ns
t_{INSU} *(3)*	3.1		3.7		4.7		ns
t_{INH} *(3)*	0.0		0.0		0.0		ns
t_{OUTCO} *(3)*	2.0	3.7	2.0	4.4	2.0	6.3	ns
t_{INSU} *(4)*	2.1		2.7		–		ns
t_{INH} *(4)*	0.0		0.0		–		ns
t_{OUTCO} *(4)*	0.5	2.7	0.5	3.4	–	–	ns
t_{PCISU}	3.0		4.2		–		ns
t_{PCIH}	0.0		0.0		–		ns
t_{PCICO}	2.0	6.0	2.0	8.9	–	–	ns

Table 75. EPF10K200S External Bidirectional Timing Parameters *Notes (1), (2)*

Symbol	Speed Grade						Unit
	-1		-2		-3		
	Min	Max	Min	Max	Min	Max	
$t_{INSUBDIR}$	2.5		3.3		4.4		ns
$t_{INHBIDIR}$	0.0		0.0		0.0		ns
$t_{OUTCOBIDIR}$ *(3)*	2.0	3.7	2.0	4.4	2.0		ns
$t_{XZBIDIR}$ *(3)*	2.0	5.2	2.0	6.1	2.0		ns
$t_{ZXBIDIR}$ *(3)*	2.0	4.7	2.0	5.6	2.0		ns
$t_{OUTCOBIDIR}$ *(4)*	0.5	2.7	0.5	3.4	–	–	ns
$t_{XZBIDIR}$ *(4)*	0.5	4.2	0.5	5.1	–	–	ns
$t_{ZXBIDIR}$ *(4)*	0.5	3.7	0.5	4.6	–	–	ns

Notes to tables:

(1) All timing parameters are described in Tables 28 and 29 in this data sheet.
(2) These parameters are specified by characterization.
(3) This parameter is measured without the use of the ClockLock or ClockBoost circuits.
(4) This parameter is measured with the use of the ClockLock or ClockBoost circuits.

Power Consumption

The supply power (P) for FLEX 10KE devices can be calculated with the following equation:

$$P = P_{INT} + P_{IO} = (I_{CCSTANDBY} + I_{CCACTIVE}) \times V_{CC} + P_{IO}$$

The $I_{CCACTIVE}$ value depends on the switching frequency and the application logic. This value is calculated based on the amount of current that each LE typically consumes. The P_{IO} value, which depends on the device output load characteristics and switching frequency, can be calculated using the guidelines given in *Application Note 74 (Evaluating Power for Altera Devices)* in this data book.

Compared to the rest of the device, the embedded array consumes a negligible amount of power. Therefore, the embedded array can be ignored when calculating supply current.

The $I_{CCACTIVE}$ value can be calculated with the following equation:

$$I_{CCACTIVE} = K \times \mathbf{f_{MAX}} \times N \times \mathbf{tog_{LC}} \times \frac{\mu A}{MHz \times LE}$$

Where:

$\mathbf{f_{MAX}}$ = Maximum operating frequency in MHz
N = Total number of LEs used in the device
$\mathbf{tog_{LC}}$ = Average percent of LEs toggling at each clock (typically 12.5%)
K = Constant

Table 76 provides the constant (K) values for FLEX 10KE devices.

Table 76. FLEX 10KE K Constant Values

Device	K Value
EPF10K30E	4.5
EPF10K50E	4.8
EPF10K50S	4.5
EPF10K100B	5.0
EPF10K100E	4.5
EPF10K130E	4.6
EPF10K200E	4.8
EPF10K200S	4.6

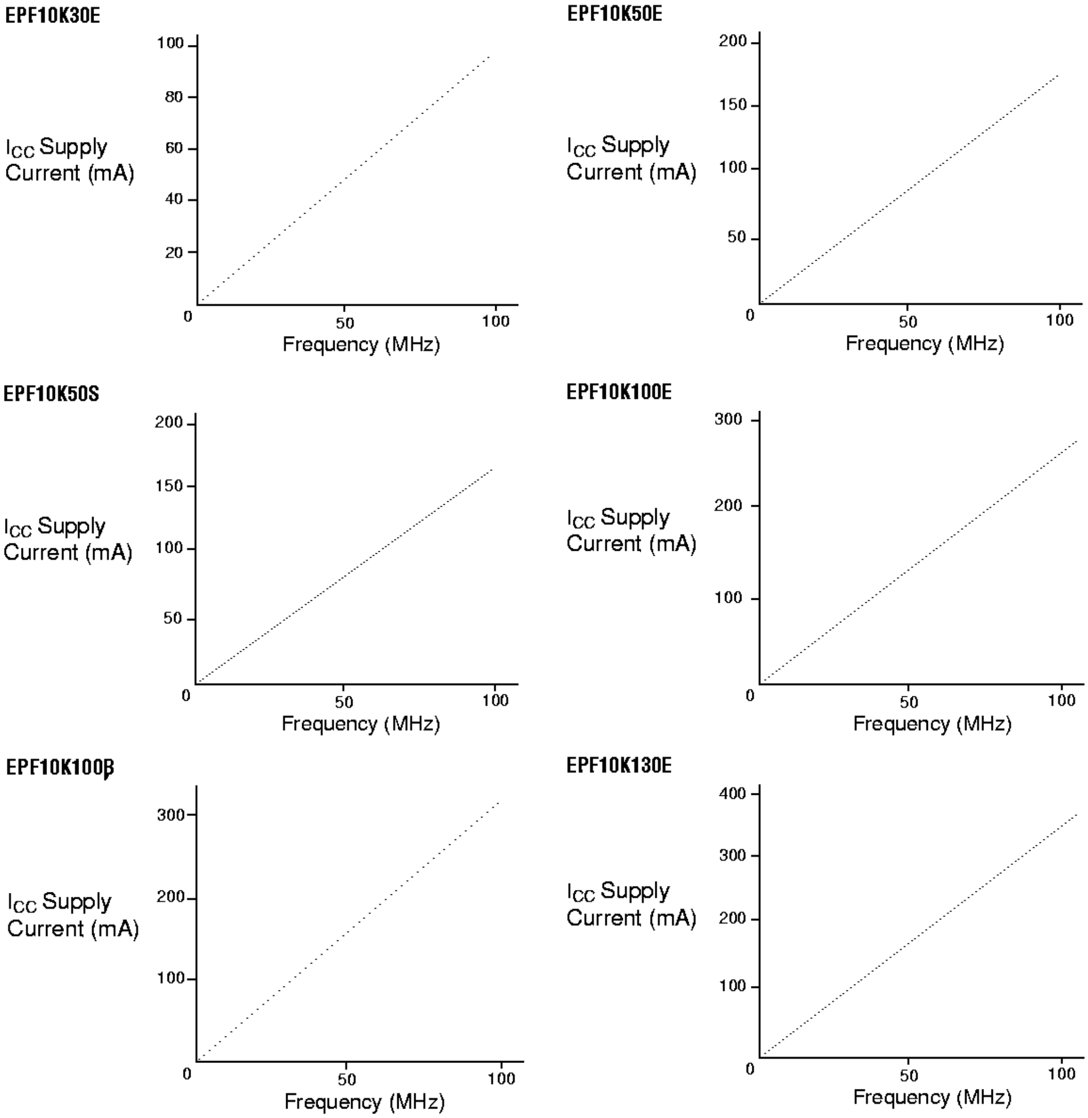
Note:
(1) Contact Altera Application for this value.

This calculation provides an I_{CC} estimate based on typical conditions with no output load. The actual I_{CC} should be verified during operation because this measurement is sensitive to the actual pattern in the device and the environmental operating conditions.

To better reflect actual designs, the power model (and the constant K in the power calculation equations) for continuous interconnect FLEX devices assumes that LEs drive FastTrack Interconnect channels. In contrast, the power model of segmented FPGAs assumes that all LEs drive only one short interconnect segment. This assumption may lead to inaccurate results when compared to measured power consumption for actual designs in segmented FPGAs.

Figure 29 shows the relationship between the current and operating frequency of FLEX 10KE devices. For information on other FLEX 10KE devices, contact Altera Applications at (800) 800-EPLD.

Figure 29. FLEX 10KE $I_{CCACTIVE}$ vs. Operating Frequency (Part 1 of 2)



EPF10K30E

EPF10K50E

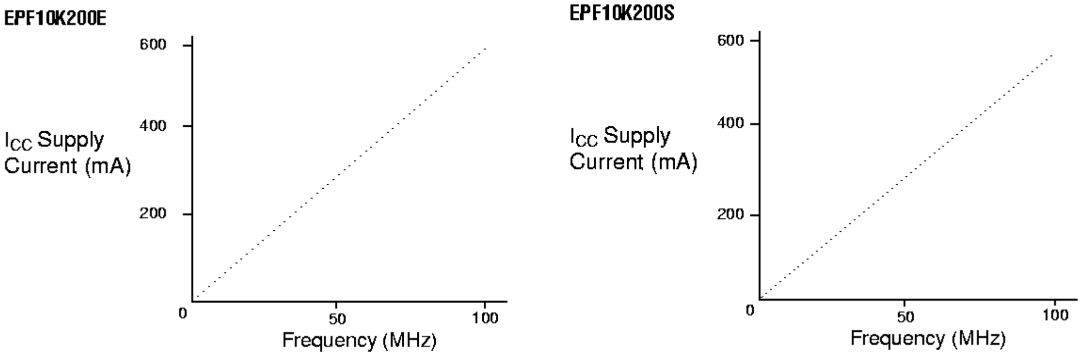
EPF10K50S

EPF10K100E

EPF10K100β

EPF10K130E

Figure 30. Figure 29. FLEX 10KE $I_{CCACTIVE}$ vs. Operating Frequency (Part 2 of 2)



Configuration & Operation

The FLEX 10KE architecture supports several configuration schemes. This section summarizes the device operating modes and available device configuration schemes.

Operating Modes

The FLEX 10KE architecture uses SRAM configuration elements that require configuration data to be loaded every time the circuit powers up. The process of physically loading the SRAM data into the device is called *configuration*. Before configuration, as V_{CC} rises, the device initiates a Power-On Reset (POR). This POR event clears the device and prepares it for configuration. The FLEX 10KE POR time does not exceed 50 µs; however, when configuring with a configuration device, the configuration device imposes a 100 ms delay that allows system power to stabilize before configuration.

During initialization, which occurs immediately after configuration, the device resets registers, enables I/O pins, and begins to operate as a logic device. The I/O pins are tri-stated during power-up, and before and during configuration. Together, the configuration and initialization processes are called *command mode*; normal device operation is called *user mode*.

SRAM configuration elements allow FLEX 10KE devices to be reconfigured in-circuit by loading new configuration data into the device. Real-time reconfiguration is performed by forcing the device into command mode with a device pin, loading different configuration data, reinitializing the device, and resuming user-mode operation. The entire reconfiguration process requires less than 276 ms and can be used to reconfigure an entire system dynamically. In-field upgrades can be performed by distributing new configuration files.

Before and during configuration, all I/O pins (except dedicated inputs, clock, or configuration pins) are pulled high by a weak pull-up resistor.

Programming Files

Despite being function- and pin-compatible, FLEX 10KE devices are not programming- or configuration file-compatible with FLEX 10K or FLEX 10KA devices. A design therefore must be recompiled before it is transferred from a FLEX 10K or FLEX 10KA device to an equivalent FLEX 10KE device. This recompilation should be performed both to create a new programming or configuration file and to check design timing in FLEX 10KE devices, which has different timing characteristics than FLEX 10K or FLEX 10KA devices.

FLEX 10KE devices are generally pin-compatible with equivalent FLEX 10KA devices. In some cases, FLEX 10KE devices have fewer I/O pins than the equivalent FLEX 10KA devices. Table 77 shows which FLEX 10KE devices have fewer I/O pins than equivalent FLEX 10KA devices. However, power, ground, JTAG, and configuration pins are the same on FLEX 10KA and FLEX 10KE devices, enabling migration from a FLEX 10KA design to a FLEX 10KE design.

Additionally, Quartus and MAX+PLUS II software offer features to help plan for future device migration by preventing the use of conflicting I/O pins.

Table 77. I/O Counts for FLEX 10KA & FLEX 10KE Devices

FLEX 10KA		FLEX 10KE	
Device	**I/O Count**	**Device**	**I/O Count**
EPF10K30AF256	191	EPF10K30EF256	176
EPF10K30AF484	246	EPF10K30EF484	220
EPF10K50VB356	274	EPF10K50EB356	220
EPF10K50VB356	274	EPF10K50SB356	220
EPF10K50VF484	291	EPF10K50EF484	254
EPF10K50VF484	291	EPF10K50SF484	254
EPF10K100AF484	369	EPF10K100EF484	338

Configuration Schemes

The configuration data for a FLEX 10KE device can be loaded with one of five configuration schemes (see Table 78), chosen on the basis of the target application. An EPC2, EPC1, or EPC1441 configuration device, intelligent controller, or the JTAG port can be used to control the configuration of a FLEX 10KE device, allowing automatic configuration on system power-up.

Multiple FLEX 10KE devices can be configured in any of the five configuration schemes by connecting the configuration enable (`nCE`) and configuration enable output (`nCEO`) pins on each device. Additional FLEX 10K, FLEX 10KA, FLEX 10KE, and FLEX 6000 devices can be configured in the same serial chain.

Table 78. Data Sources for FLEX 10KE Configuration

Configuration Scheme	Data Source
Configuration device	EPC1, EPC2, or EPC1441 configuration device
Passive serial (PS)	BitBlaster, ByteBlaster, or ByteBlasterMV download cables, or serial data source
Passive parallel asynchronous (PPA)	Parallel data source
Passive parallel synchronous (PPS)	Parallel data source
JTAG	BitBlaster or ByteBlasterMV download cables, or microprocessor with a Jam File or JBC File

Device Pin-Outs

Tables 79 and 80 shows the dedicated pin-outs for FLEX 10KE devices in 144-pin TQFP, 208-pin PQFP, 240-pin PQFP, 356-pin BGA, 599-pin PGA, and 600-pin BGA packages.

Table 79. FLEX 10KE Device Pin-Outs (Part 1 of 2) *Note (1)*

Pin Name	144-Pin TQFP EPF10K30E EPF10K50E EPF10K50S	208-Pin PQFP EPF10K30E EPF10K50E EPF10K50S EPF10K100E EPF10K100β	240-Pin PQFP *(2), (3)* EPF10K50E EPF10K50S EPF10K100E EPF10K100β	240-Pin PQFP *(2)* EPF10K130E	240-Pin RQFP *(3)* EPF10K200S
`MSEL0` *(4)*	77	108	124	124	124
`MSEL1` *(4)*	76	107	123	123	123
`nSTATUS` *(4)*	35	52	60	60	60
`nCONFIG` *(4)*	74	105	121	121	121
`DCLK` *(4)*	107	155	179	179	179
`CONF_DONE` *(4)*	2	2	2	2	2
`INIT_DONE` *(5)*	14	19	26	26	26
`nCE` *(4)*	106	154	178	178	178
`nCEO` *(6)*	3	3	3	3	3
`nWS` *(6)*	142	206	238	238	238
`nRS` *(6)*	141	204	236	236	236
`nCS` *(6)*	144	208	240	240	240
`CS` *(6)*	143	207	239	239	239
`RDYnBUSY` *(6)*	11	16	23	23	23
`CLKUSR` *(6)*	7	10	11	11	11
`DATA7` *(6)*	116	166	190	190	190
`DATA6` *(6)*	114	164	188	188	188
`DATA5` *(6)*	113	162	186	186	186
`DATA4` *(6)*	112	161	185	185	185
`DATA3` *(6)*	111	159	183	183	183
`DATA2` *(6)*	110	158	182	182	182
`DATA1` *(6)*	109	157	181	181	181
`DATA0` *(4), (7)*	108	156	180	180	180
`TDI` *(4)*	105	153	177	177	177
`TDO` *(4)*	4	4	4	4	4
`TCK` *(4)*	1	1	1	1	1
`TMS` *(4)*	34	50	58	58	58
`TRST` *(4)*	*(8)*	51	59	59	59

Table 79. FLEX 10KE Device Pin-Outs (Part 2 of 2) *Note (1)*

Pin Name	**144-Pin TQFP** **EPF10K30E** **EPF10K50E** **EPF10K50S**	**208-Pin** **PQFP** **EPF10K30E** **EPF10K50E** **EPF10K50S** **EPF10K100E** **EPF10K100β**	**240-Pin** **PQFP** *(2), (3)* **EPF10K50E** **EPF10K50S** **EPF10K100E** **EPF10K100β**	**240-Pin** **PQFP** *(2)* **EPF10K130E**	**240-Pin** **RQFP** *(3)* **EPF10K200S**
Dedicated Inputs	54, 56, 124, 126	78, 80, 182, 184	90, 92, 210, 212	90, 92, 210, 212	90, 92, 210, 212
Dedicated Clock Pins	55, 125	79, 183	91, 211	91, 211	91, 211
`GCLK1` *(9)*	55	79	91	91	91
Lock *(10)*	42	62	73	73	73
`DEV_CLRn` *(5)*	122	180	209	209	209
`DEV_OE` *(5)*	128	186	213	213	213
`VCCINT` (2.5 V)	6, 25, 52, 75, 93, 123	6, 23, 35, 43, 76, 106, 109, 117, 137, 145, 181	5, 27, 47, 96, 122, 130, 150, 170	5, 20, 27, 47, 76, 96, 122, 130, 150, 159, 170	5, 20, 27, 40, 47, 76, 96, 122, 130, 139, 150, 159, 170, 187, 225
`VCCIO` (2.5 or 3.3 V)	5, 24, 45, 61, 71, 94, 115, 134	5, 22, 34, 42, 66, 84, 98, 110, 118, 138, 146, 165, 178, 194	16, 37, 57, 77, 112, 140, 160, 189, 205, 224	16, 37, 57, 77, 112, 140, 160, 189, 205, 224	16, 37, 57, 77, 112, 140, 160, 189, 205, 224
`VCC_CKLK` *(11)*	53	77 *(12)*	89 *(12)*	89	89
`GNDINT`	15, 16, 40, 50, 58, 66, 84, 85, 103, 104, 127, 129, 139	20, 21, 32, 33, 48, 49, 59, 72, 82, 91, 123, 124, 129, 130, 151, 152, 171, 185, 188, 201	10, 22, 32, 42, 52, 69, 85, 104, 125, 135, 145, 155, 165, 176, 197, 216, 232	10, 22, 32, 42, 52, 69, 85, 104, 125, 135, 145, 155, 165, 176, 197, 216, 232	10, 22, 32, 42, 52, 69, 85, 104, 125, 135, 145, 155, 165, 176, 197, 216, 232
`GNDIO`	–	–	–	–	–
`GND_CKLK` *(11)*	57	81 *(13)*	93 *(13)*	93	93
Total User I/O Pins *(14)*	102	147	189	186	182

Table 80. FLEX 10KE Device Pin-Outs (Part 1 of 4) *Note (1)*

Pin Name	356-Pin βGA EPF10K50E EPF10K50S	356-Pin βGA EPF10K100E EPF10K130E EPF10K200S	599-Pin PGA EPF10K200E EPF10K200S	600-Pin βGA EPF10K130E	600-Pin βGA EPF10K200E EPF10K200S
`MSEL0` *(4)*	D4	D4	F6	F5	F5
`MSEL1` *(4)*	D3	D3	C3	C1	C1
`nSTATUS` *(4)*	D24	D24	E43	D32	D32
`nCONFIG` *(4)*	D2	D2	B4	D4	D4
`DCLK` *(4)*	AC5	AC5	BE5	AP1	AP1
`CONF_DONE` *(4)*	AC24	AC24	BC43	AM32	AM32
`INIT_DONE` *(5)*	T24	T24	AM40	AE32	AE32
`nCE` *(4)*	AC2	AC2	BB6	AN2	AN2
`nCEO` *(6)*	AC22	AC22	BF44	AP35	AP35
`nWS` *(6)*	AE24	AE24	BB40	AR29	AR29
`nRS` *(6)*	AE23	AE23	BA37	AM28	AM28
`nCS` *(6)*	AD24	AD24	AY38	AL29	AL29
`CS` *(6)*	AD23	AD23	BA39	AN29	AN29
`RDYnBUSY` *(6)*	U22	U22	AW47	AG35	AG35
`CLKUSR` *(6)*	AA24	AA24	AY42	AM34	AM34
`DATA7` *(6)*	AF4	AF4	BD14	AM13	AM13
`DATA6` *(6)*	AD8	AD8	BA17	AR12	AR12
`DATA5` *(6)*	AE5	AE5	BB16	AN12	AN12
`DATA4` *(6)*	AD6	AD6	BF12	AP11	AP11
`DATA3` *(6)*	AF2	AF2	BG11	AM11	AM11
`DATA2` *(6)*	AD5	AD5	BG9	AR10	AR10
`DATA1` *(6)*	AD4	AD4	BF10	AN10	AN10
`DATA0` *(4), (7)*	AD3	AD3	BC5	AM4	AM4
`TDI` *(4)*	AC3	AC3	BF4	AN1	AN1
`TDO` *(4)*	AC23	AC23	BB42	AN34	AN34
`TCK` *(4)*	AD25	AD25	BE43	AL31	AL31
`TMS` *(4)*	D22	D22	F42	C35	C35
`TRST` *(4)*	D23	D23	B46	C34	C34
Dedicated Inputs	A13, B14, AF14, AE13	A13, B14, AF14, AE13	B24, C25, BG25, BG23	C18, D18, AM18, AN18	C18, D18, AM18, AN18
Dedicated Clock Pins	A14, AF13	A14, AF13	BF24, A25	AL18, E18	AL18, E18
`GCLK1` *(9)*	A14	A14	–	E18	E18
Lock *(10)*	C18	C18	–	A23	A23

Table 80. FLEX 10KE Device Pin-Outs (Part 2 of 4) *Note (1)*

Pin Name	**356-Pin** **βGA** **EPF10K50E** **EPF10K50S**	**356-Pin** **βGA** **EPF10K100E** **EPF10K130E** **EPF10K200S**	**599-Pin** **PGA** **EPF10K200E** **EPF10K200S**	**600-Pin** **βGA** **EPF10K130E**	**600-Pin** **βGA** **EPF10K200E** **EPF10K200S**
`DEV_CLRn` *(5)*	AD13	AD13	BE23	AR17	AR17
`DEV_OE` *(5)*	AE14	AE14	BC25	AR19	AR19
`VCCINT` (2.5 V)	A1, A26, C26, D5, F1, H22, J1, M26, N1, T26, U5, AA1, AD26, AF1, AF26	A1, A26, C26, D5, F1, H22, J1, M26, N1, T26, U5, AA1, AD26, AF1, AF26	A3, A45, C1, C11, C19, C29, C37, C47, E5, G25, L3, L45, W3, W45, AJ3, AJ45, AU3, AU45, BE1, BE11, BE19, BE29, BE37, BE47, BG3, BG45	A11, A19, B1, D24, E2, F31, F35, H1, K32, M2, N34, P5, T35, U3, V32, Y2, AA33, AB5, AD35, AE4, AF32, AG5, AK31, AK35, AL3, AP24, AR11, AR18	A11, A19, B1, D24, E2, F31, F35, H1, K32, M2, N34, P5, T35, U3, V32, Y2, AA33, AB5, AD35, AE4, AF32, AG5, AK31, AK35, AL3, AP24, AR11, AR18
`VCCIO` (2.5 or 3.3 V)	A7, A23, B4, C15, D25, F4, H24, K5, M23, P2, T25, V2, W22, AB1, AC25, AD18, AF3, AF7, AF16	A7, A23, B4, C15, D25, F4, H24, K5, M23, P2, T25, V2, W22, AB1, AC25, AD18, AF3, AF7, AF16	D24, E9, E15, E21, E27, E33, E39, G7, G41, J5, J43, R5, R43, AA5, AA43, AD4, AD44, AG5, AG43, AN5, AN43, AW5, AW43, BA7, BA41, BC9, BC15, BC21, BC27, BC33, BC39, BD24	C8, E12, C15, A20, C23, A27, AM26, AR23, AM19, AN15, AL12, AN8, C2, C3, C4, D5, E5, C33, C32, D31, E31, AL5, AM5, AN4, AN3, AM31, AN32, AN33, AP34	C8, E12, C15, A20, C23, A27, AM26, AR23, AM19, AN15, AL12, AN8, C2, C3, C4, D5, E5, C33, C32, D31, E31, AL5, AM5, AN4, AN3, AM31, AN32, AN33, AP34
`VCC_CKLK` *(11)*	C14	C14	–	B18	B18

Table 80. FLEX 10KE Device Pin-Outs (Part 3 of 4) *Note (1)*

Pin Name	**356-Pin BGA EPF10K50E EPF10K50S**	**356-Pin BGA EPF10K100E EPF10K130E EPF10K200S**	**599-Pin PGA EPF10K200E EPF10K200S**	**600-Pin BGA EPF10K130E**	**600-Pin BGA EPF10K200E EPF10K200S**
`GNDINT`	A2, A10, A20, B1, B22, B25, B26, C2, C9, C13, C25, H23, J26, K1, M1, N26, R1, R26, T1, U26, W1, AD2, AD14, AD20, AE1, AE2, AE7, AE25, AE26, AF11, AF19, AF25	A2, A10, A20, B1, B22, B25, B26, C2, C9, C13, C25, H23, J26, K1, M1, N26, R1, R26, T1, U26, W1, AD2, AD14, AD20, AE1, AE2, AE7, AE25, AE26, AF11, AF19, AF25	A47, B2, C13, C21, C27, C35, C45, D4, G23, N3, N45, AA3, AA45, AG3, AG45, AR3, AR45, BD44, BE3, BE13, BE21, BE27, BE35, BE45, BG1, BG47	A1, A2, A3, A4, A5, , A31, A32, A33, A34, A35, B2, B3, B4, B5, B6, B31, B32, B33, B34, B35, C5, C6, D6, E6, C30, C31, D30, E30, AL6, AL30, AM6, AM30, AN5, AN6, AN30, AN31, AN35, AP2, AP3, AP4, AP5, AP6, AP30, AP31, AP32, AP33, AR1, AR2, AR3, AR4, AR5, AR30, AR31, AR32, AR33, AR34, AR35	A1, A2, A3, A4, A5, , A31, A32, A33, A34, A35, B2, B3, B4, B5, B6, B31, B32, B33, B34, B35, C5, C6, D6, E6, C30, C31, D30, AN35
`GNDIO`	–	–	E7, E13, E19, E29, E35, E41, F24, G5, G43, H40, N5, N43, W5, W43, AD6, AD42, AJ5, AJ43, AR5, AR43, AY8, AY40, BA5, BA43, BB24, BC7, BC13, BC19, BC29, BC35, BC41	–	E30, AL6, AL30, AM6, AM30, AN5, AN6, AN30, AN31, AP2, AP3, AP4, AP5, AP6, AP30, AP31, AP32, AP33, AR1, AR2, AR3, AR4, AR5, AR30, AR31, AR32, AR33, AR34, AR35
`GND_CKLK` *(11)*	B13	B13	–	A18	A18

Table 80. FLEX 10KE Device Pin-Outs (Part 4 of 4) *Note (1)*

Pin Name	356-Pin βGA EPF10K50E EPF10K50S	356-Pin βGA EPF10K100E EPF10K130E EPF10K200S	599-Pin PGA EPF10K200E EPF10K200S	600-Pin βGA EPF10K130E	600-Pin βGA EPF10K200E EPF10K200S
No Connect (N.C.)	D1, E2, E22, E25, F5, F23, F26, G3, G22, H4, H5, J2, J4, J23, J25, K2, K3, K25, K26, L2, L23, L26, M2, M5, M22, M25, N4, N25, P1, P5, P22, P23, R5, T22, U2, U3, U23, U24, V4, W3, W4, W24, W26, Y2, Y5, AA3, AA22, AA25, AB3, AB5, AB22, AB24, AB26	–	–	D3, D35, E1, F34, G2, H5, J3, J4, J32, K1, L4, L31, M3, N1, N33, N35, P4, P33, R2, R32, T4, U5, U34, V3, V34, W1, W32, W35, Y31, AA2, AA34, AB1, AB31, AB34, AB35, AC31, AC34, AE33, AE35, AF1, AG3, AH2, AJ32, AK2, AK32, AL33	–
Total User I/O Pins *(14)*	220	274	470	424	470

Notes to tables:

(1) All pins that are not listed are user I/O pins.
(2) EPF10K50E, EPF10K100E, and EPF10K100B devices are pin-compatible with the EPF10K130E devices in the same package if pins 20, 76, and 159 are connected to `VCCINT`. The Quartus and MAX+PLUS II software performs this function automatically when future migration is set.
(3) EPF10K50E, EPF10K100E, and EPF10K100B devices are pin-compatible with the EPF10K200E devices in the same package if pins 20, 40, 76, 139, 159, 187, and 225 are connected to `VCCINT`. The Quartus and MAX+PLUS II software performs this function automatically when future migration is set.
(4) This pin is a dedicated pin; it is not available as a user I/O pin.
(5) This pin can be used as a user I/O pin if it is not used for its device-wide or configuration function.
(6) This pin can be used as a user I/O pin after configuration.
(7) This pin is tri-stated in user mode.
(8) The optional JTAG pin `TRST` is not used in the 144-pin TQFP package.
(9) This pin drives the ClockLock and ClockBoost circuitry.
(10) This pin shows the status of the ClockLock and ClockBoost circuitry. When the ClockLock and ClockBoost circuitry is locked to the incoming clock and generates an internal clock, `LOCK` is driven high. `LOCK` remains high if a periodic clock stops clocking. The `LOCK` function is optional; if the `LOCK` output is not used, this pin is a user I/O pin.
(11) This pin is the power or ground for the ClockLock and ClockBoost circuitry. To ensure noise resistance, the power and ground supply to the ClockLock and ClockBoost circuitry should be isolated from the power and ground to the rest of the device. If the ClockLock or ClockBoost circuitry is not used, this power or ground pin should be connected to `VCCINT` or `GNDINT`, respectively.
(12) When using the EPF10K100B device, connect this pin to `VCCINT`.
(13) When using the EPF10K100B device, connect this pin to `GNDINT`.
(14) The user I/O pin count includes dedicated input pins, dedicated clock pins, and all I/O pins.

Tables 81 through 83 show the dedicated pin-outs for FLEX 10KE devices in 256-pin FineLine BGA, 484-pin FineLine BGA, and 672-pin FineLine BGA packages.

Table 81. FLEX 10KE FineLine BGA Device Pin-Outs (Part 1 of 4) *Notes (1), (2)*

Pin Name	256-Pin FineLine BGA EPF10K30E	256-Pin FineLine BGA EPF10K50E EPF10K50S EPF10K100E EPF10K100B	484-Pin FineLine BGA EPF10K30E
MSEL0 *(3)*	P1	P1	U4
MSEL1 *(3)*	R1	R1	V4
nSTATUS *(3)*	T16	T16	W19
nCONFIG *(3)*	N4	N4	T7
DCLK *(3)*	B2	B2	E5
CONF_DONE *(3)*	C15	C15	F18
INIT_DONE *(4)*	G16	G16	K19
nCE *(3)*	B1	B1	E4
nCEO *(3)*	B16	B16	E19
nWS *(5)*	B14	B14	E17
nRS *(5)*	C14	C14	F17

Table 81. FLEX 10KE FineLine βGA Device Pin-Outs (Part 2 of 4) *Notes (1), (2)*

Pin Name	256-Pin FineLine βGA EPF10K30E	256-Pin FineLine βGA EPF10K50E EPF10K50S EPF10K100E EPF10K100β	484-Pin FineLine βGA EPF10K30E
`nCS` *(5)*	A16	A16	D19
`CS` *(5)*	A15	A15	D18
`RDYnBSY` *(5)*	G14	G14	K17
`CLKUSR` *(5)*	D15	D15	G18
`DATA7` *(5)*	B5	B5	E8
`DATA6` *(5)*	D4	D4	G7
`DATA5` *(5)*	A4	A4	D7
`DATA4` *(5)*	B4	B4	E7
`DATA3` *(5)*	C3	C3	F6
`DATA2` *(5)*	A2	A2	D5
`DATA1` *(5)*	B3	B3	E6
`DATA0` *(3), (6)*	A1	A1	D4
`TDI` *(3)*	C2	C2	F5
`TDO` *(3)*	C16	C16	F19
`TCK` *(3)*	B15	B15	E18
`TMS` *(3)*	P15	P15	U18
`TRST` *(3)*	R16	R16	V19
Dedicated Inputs	B9, E8, M9, R8	B9, E8, M9, R8	E12, H11, R12, V11
Dedicated Clock Pins	A9, L8	A9, L8	D12, P11
`GCLK1` *(7)*	L8	L8	P11
`LOCK` *(8)*	P12	P12	U15
`DEV_CLRn` *(4)*	D8	D8	G11
`DEV_OE` *(4)*	C9	C9	F12
`VCCINT` (2.5 V)	E11, F5, F7, F9, F12, H6, H7, H10, J7, J10, J11, K9, L5, L7, L12, M11, R2	E11, F5, F7, F9, F12, H6, H7, H10, J7, J10, J11, K9, L5, L7, L12, M11, R2	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P15, R14, V5, W21, Y8, AA12
`VCCIO` (2.5 or 3.3 V)	D12, E6, F8, F10, G6, G8, G11, H11, J6, K6, K8, K11, L10, M6, N12	D12, E6, F8, F10, G6, G8, G11, H11, J6, K6, K8, K11, L10, M6, N12	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13

Table 81. FLEX 10KE FineLine βGA Device Pin-Outs (Part 3 of 4) Notes (1), (2)			
Pin Name	**256-Pin FineLine βGA EPF10K30E**	**256-Pin FineLine βGA EPF10K50E EPF10K50S EPF10K100E EPF10K100β**	**484-Pin FineLine βGA EPF10K30E**
`VCC_CKLK` *(9)*	L9	L9 *(10)*	P12
`GND`	E5, E12, F6, F11, G7, G9, G10, H8, H9, J8, J9, K7, K10, L6, L11, M5, M12	E5, E12, F6, F11, G7, G9, G10, H8, H9, J8, J9, K7, K10, L6, L11, M5, M12	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16
`GND_CKLK` *(9)*	T8	T8 *(11)*	W11

Table 81. FLEX 10KE FineLine βGA Device Pin-Outs (Part 4 of 4) *Notes (1), (2)*

Pin Name	**256-Pin FineLine βGA EPF10K30E**	**256-Pin FineLine βGA** **EPF10K50E** **EPF10K50S** **EPF10K100E** **EPF10K100β**	**484-Pin FineLine βGA EPF10K30E**
No Connect (N.C.)	D1, E3, E16, G3, H1, H16, J1, K3, K14, K16, L2, L4, M14, M16, N15	–	A2, A3, A4, A5, A7, A9, A11, A12, A14, A15, A20, A21, B3, B4, B9, B10, B12, B16, B19, B20, C1, C6, C9, C10, C12, C13, C14, C16, C17, C22, D1, D2, D3, D20, D21, D22, E2, E3, E20, E22, F1, F2, F3, F20, F21, F22, G2, G4, G20, G22, H1, H3, H6, H19, H21, H22, J1, J2, J3, J21, J22, K1, K2, K6, K21, K22, L1, L2, L3, L4, L19, L20, L21, L22, M1, M2, M3, M4, M21, M22, N1, N2, N21, N22, N6, N17, N19, P1, P2, P3, P5, P7, P20, P21, P22, R2, R3, R17, R19, R20, R21, T2, T18, T20, T21, U1, U2, U3, U20, U21, U22, V1, V2, V20, V21, W1, W2, W22, Y4, Y9, Y12, Y13, Y16, Y19, Y20, AA2, AA3, AA4, AA9, AA11, AA13, AA15, AA21, AB1, AB2, AB3, AB4, AB5, AB7, AB8, AB9, AB12, AB15, AB17, AB18, AB19, AB20, AB21, AB22
Total User I/O Pins *(12)*	176	191	220

Table 82. FLEX 10KE FineLine βGA Device Pin-Outs (Part 1 of 2) *Notes (1), (2)*

Pin Name	484-Pin FineLine βGA EPF10K50E EPF10K50S	484-Pin FineLine βGA EPF10K100E
`MSEL0` *(3)*	U4	U4
`MSEL1` *(3)*	V4	V4
`nSTATUS` *(3)*	W19	W19
`nCONFIG` *(3)*	T7	T7
`DCLK` *(3)*	E5	E5
`CONF_DONE` *(3)*	F18	F18
`INIT_DONE` *(4)*	K19	K19
`nCE` *(3)*	E4	E4
`nCEO` *(3)*	E19	E19
`nWS` *(5)*	E17	E17
`nRS` *(5)*	F17	F17
`nCS` *(5)*	D19	D19
`CS` *(5)*	D18	D18
`RDYnBSY` *(5)*	K17	K17
`CLKUSR` *(5)*	G18	G18
`DATA7` *(5)*	E8	E8
`DATA6` *(5)*	G7	G7
`DATA5` *(5)*	D7	D7
`DATA4` *(5)*	E7	E7
`DATA3` *(5)*	F6	F6
`DATA2` *(5)*	D5	D5
`DATA1` *(5)*	E6	E6
`DATA0` *(3)*, *(6)*	D4	D4
`TDI` *(3)*	F5	F5
`TDO` *(3)*	F19	F19
`TCK` *(3)*	E18	E18
`TMS` *(3)*	U18	U18
`TRST` *(3)*	V19	V19
Dedicated Inputs	E12, H11, R12, V11	E12, H11, R12, V11
Dedicated Clock Pins	D12, P11	D12, P11
`GCLK1` *(7)*	P11	P11
`LOCK` *(8)*	U15	U15
`DEV_CLRn` *(4)*	G11	G11
`DEV_OE` *(4)*	F12	F12

Table 82. FLEX 10KE FineLine βGA Device Pin-Outs (Part 2 of 2) *Notes (1), (2)*

Pin Name	484-Pin FineLine βGA EPF10K50E EPF10K50S	484-Pin FineLine βGA EPF10K100E
`VCCINT` (2.5 V)	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P15, R14, V5, W21, Y8, AA12	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P15, R14, V5, W21, Y8, AA12
`VCCIO` (2.5 or 3.3 V)	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13
`VCC_CKLK` *(9)*	P12	P12
`GND`	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16
`GND_CKLK` *(9)*	W11	W11
No Connect (N.C.)	A2, A3, A4, A5, A7, A9, A11, A12, A14, A15, A20, A21, B3, B4, B9, B10, B12, B16, B19, B20, C1, C6, C9, C10, C12, C13, C14, C16, C17, C22, D1, D2, D3, D20, D21, E2, E3, E20, E22, F1, F2, F20, F21, G2, G20, G22, J1, J2, J3, J21, K2, K22, L1, L2, L20, L22, M2, M3, M22, N1, N2, N21, N22, P3, P20, P21, P22, R2, R3, R21, T2, T20, T21, U1, U2, U3, U20, U21, U22, V2, V20, W1, W2, W22, Y4, Y9, Y12, Y13, Y16, Y19, Y20, AA2, AA3, AA4, AA9, AA11, AA13, AA15, AA21, AB1, AB2, AB3, AB4, AB5, AB7, AB8, AB9, AB12, AB15, AB17, AB18, AB19, AB20, AB21, AB22	A2, A3, A4, A5, B3, B4, B10, C17, F2, J2, K2, L2, N1, P20, P22, R3, T20, T21, U1, W22, Y16, AA15, AB3, AB4, AB5, AB7, AB15, AB17, AB18, AB19, AB20
Total User I/O Pins *(12)*	254	338

Table 83. FLEX 10KE FineLine βGA Device Pin-Outs (Part 1 of 3) *Notes (1), (2)*

Pin Name	484-Pin FineLine βGA EPF10K130E EPF10K200S	672-Pin FineLine βGA EPF10K130E	672-Pin FineLine βGA EPF10K200E EPF10K200S
`MSEL0` *(3)*	U4	W6	W6
`MSEL1` *(3)*	V4	Y6	Y6
`nSTATUS` *(3)*	W19	AA21	AA21
`nCONFIG` *(3)*	T7	V9	V9
`DCLK` *(3)*	E5	G7	G7
`CONF_DONE` *(3)*	F18	H20	H20
`INIT_DONE` *(4)*	K19	M21	M21
`nCE` *(3)*	E4	G6	G6
`nCEO` *(3)*	E19	G21	G21
`nWS` *(5)*	E17	G19	G19
`nRS` *(5)*	F17	H19	H19
`nCS` *(5)*	D19	F21	F21
`CS` *(5)*	D18	F20	F20
`RDYnBSY` *(5)*	K17	M19	M19
`CLKUSR` *(5)*	G18	J20	J20
`DATA7` *(5)*	E8	G10	G10
`DATA6` *(5)*	G7	J9	J9
`DATA5` *(5)*	D7	F9	F9
`DATA4` *(5)*	E7	G9	G9
`DATA3` *(5)*	F6	H8	H8
`DATA2` *(5)*	D5	F7	F7
`DATA1` *(5)*	E6	G8	G8
`DATA0` *(3), (6)*	D4	F6	F6
`TDI` *(3)*	F5	H7	H7
`TDO` *(3)*	F19	H21	H21
`TCK` *(3)*	E18	G20	G20
`TMS` *(3)*	U18	W20	W20
`TRST` *(3)*	V19	Y21	Y21
Dedicated Inputs	E12, H11, R12, V11	Y13, U14, G14, K13	Y13, U14, G14, K13
Dedicated Clock Pins	D12, P11	T13, F14	T13, F14
`GCLK1` *(7)*	P11	T13	T13
`LOCK` *(8)*	U15	W17	W17
`DEV_CLRn` *(4)*	G11	J13	J13
`DEV_OE` *(4)*	F12	H14	H14

Table 83. FLEX 10KE FineLine βGA Device Pin-Outs (Part 2 of 3) *Notes (1), (2)*

Pin Name	**484-Pin FineLine βGA EPF10K130E EPF10K200S**	**672-Pin FineLine βGA EPF10K130E**	**672-Pin FineLine βGA EPF10K200E EPF10K200S**
`VCCINT` (2.5 V)	C11, C15, H14, J8, J10, J12, J15, L9, L10, L13, M10, M13, M14, N12, P8, P10, P15, R14, V5, W21, Y8, AA12	E13, E17, H2, H25, K16, L10, L12, L14, L17, M2, M25, N11, N12, N15, P12, P15, P16, R14, T2, T10,T12, T17, T25, U16, Y7, AA23, AB10, AC14	E13, E17, H2, H25, K16, L10, L12, L14, L17, M2, M25, N11, N12, N15, P12, P15, P16, R14, T2, T10,T12, T17, T25, U16, Y7, AA23, AB10, AC14
`VCCIO` (2.5 or 3.3 V)	A6, A13, B5, E1, G1, G15, H9, H20, J11, J13, K9, K11, K14, K20, L14, M9, N3, N9, N11, N14, N20, P13, R1, R9, T3, T15, T22, V22, AB13	C8, C15, D7, G3, J3, J17, K11, K22, L13, L15, M11, M13, M16, M22, N16, P11, R5, R11, R13, R16, R22, T15, U3, U11, V5, V17, V24, Y2, Y24, AA26, AD15	C8, C15, D7, G3, J3, J17, K11, K22, L13, L15, M11, M13, M16, M22, N16, P11, R5, R11, R13, R16, R22, T15, U3, U11, V5, V17, V24, Y2, Y24, AA26, AD15
`VCC_CKLK` *(9)*	P12	T14	T14
`GND`	A1, A8, A22, B1, B2, B17, B21, B22, C2, C21, E21, G3, G21, H2, H8, H15, J9, J14, J20, K3, K10, K12, K13, L11, L12, M11, M12, M20, N10, N13, P9, P14, R8, R15, R22, T1, V3, W20, Y1, Y2, Y3, Y21, Y22, AA1, AA6, AA22, AB11, AB16	A2, A25, B2, B25, C3, C10, C24, D3, D4, D19, D23, D24, E4, E23, G23, J5, J23, K4, K10, K17, L11, L16, L22, M5, M12, M14, M15, N13, N14, P13, P14, P22, R12, R15, T11, T16, U10, U17, U24, V3, Y5, AA22, AB3, AB4, AB5, AB23, AB24, AC3, AC8, AC24, AD13, AD18, AE2, AE25, AF2, AF25	A2, A25, B2, B25, C3, C10, C24, D3, D4, D19, D23, D24, E4, E23, G23, J5, J23, K4, K10, K17, L11, L16, L22, M5, M12, M14, M15, N13, N14, P13, P14, P22, R12, R15, T11, T16, U10, U17, U24, V3, Y5, AA22, AB3, AB4, AB5, AB23, AB24, AC3, AC8, AC24, AD13, AD18, AE2, AE25, AF2, AF25
`GND_CKLK` *(9)*	W11	AA13	AA13

Table 83. FLEX 10KE FineLine βGA Device Pin-Outs (Part 3 of 3) *Notes (1), (2)*

Pin Name	484-Pin FineLine βGA EPF10K130E EPF10K200S	672-Pin FineLine βGA EPF10K130E	672-Pin FineLine βGA EPF10K200E EPF10K200S
No Connect (N.C.)	–	A3, A4, A5, A6, A8, A9, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20, A21, A22, A23, A24, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13, B14, B16, B19, B20, B21, B22, B23, B24, B26, C1, C25, C26, D1, D2, D25, D26, E1, E25, E26, F1, F25, G25, G26, H1, J1, J25, J26, K26, L2, L25, N2, P1, P2, R1, R26, T1, U1, U25, V1, V26, W1, Y26, AA1, AA2, AA25, AB2, AB25, AB26, AC1, AC2, AC25, AC26, AD2, AD26, AE1, AE3, AE4, AE5, AE6, AE7, AE8, AE9, AE10, AE11, AE12, AE14, AE15, AE16, AE17, AE19, AE20, AE21, AE22, AE23, AE24, AE26, AF3, AF4, AF5, AF6, AF7, AF8, AF9, AF10, AF11, AF12, AF13, AF14, AF15, AF16, AF17, AF18, AF20, AF21, AF23, AF24	A4, A5, A6, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20, A21, A22, A23, A24, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13, B16, B19, B20, B21, B22, B23, B24, C1, AE4, AE5, AE6, AE7, AE8, AE9, AE10, AE11, AE12, AE14, AE15, AE16, AE17, AE19, AE20, AE21, AE22, AE23, AF4, AF5, AF6, AF7, AF8, AF9, AF10, AF12, AF13, AF14, AF15, AF16, AF18, AF20, AF21, AF23, AF24
Total User I/O Pins *(12)*	369	413	470

Notes to tables:

(1) All pins that are not listed are user I/O pins.
(2) All FineLine BGA packages support SameFrame pin migration to allow migration from one package to another. The Quartus and MAX+PLUS II software performs this function automatically when future migration is set.
(3) This pin is a dedicated pin and is not available as a user I/O pin.
(4) This pin can be used as a user I/O pin if it is not used for its device-wide or configuration function.
(5) This pin can be used as a user I/O pin after configuration.
(6) This pin is tri-stated in user mode.
(7) This pin drives the ClockLock and ClockBoost circuitry.
(8) This pin shows the status of the ClockLock and ClockBoost circuitry. When the ClockLock and ClockBoost circuitry is locked to the incoming clock and generates an internal clock, `LOCK` is driven high. `LOCK` remains high if a periodic clock stops clocking. The `LOCK` function is optional; if the `LOCK` output is not used, this pin is a user I/O pin.
(9) This pin is the power or ground for the ClockLock and ClockBoost circuitry. To ensure noise resistance, the power and ground supply to the ClockLock and ClockBoost circuitry should be isolated from the power and ground to the rest of the device. If the ClockLock or ClockBoost circuitry is not used, this power or ground pin should be connected to `VCCINT` or `GND`, respectively.
(10) When using the EPF10K100B device, connect this pin to `VCCINT`.
(11) When using the EPF10K100B device, connect this pin to `GNDINT`.
(12) The user I/O pin count includes dedicated input pins, dedicated clock pins, and all I/O pins.

Table 84 shows pin compatibility between different FLEX 10KE devices.

Table 84. FLEX 10KE Device Pin Compatibility *Note (1)*

Device	144-Pin TQFP	208-Pin PQFP	240-Pin PQFP RQFP	599-Pin PGA	356-Pin βGA	600-Pin βGA	256-Pin FineLine βGA	484-Pin FineLine βGA	672-Pin FineLine βGA
EPF10K30E	*(2)*	*(2)*					*(3)*	*(3)*	*(4)*
EPF10K50E	*(2)*	*(2)*	*(3)*		*(3)*		*(3)*	*(3)*	*(4)*
EPF10K50S	*(2)*	*(2)*	*(3)*		*(3)*		*(3)*	*(3)*	*(4)*
EPF10K100B		*(2)*	*(3)*				*(3)*		
EPF10K100E		*(2)*	*(3)*		*(3)*		*(3)*	*(3)*	*(4)*
EPF10K130E			*(3)*		*(3)*	*(3)*		*(3)*	*(3)*
EPF10K200E				*(2)*		*(3)*			*(3)*
EPF10K200S			*(3)*	*(2)*	*(3)*	*(3)*		*(3)*	*(3)*

Notes:

(1) All FineLine BGA packages support SameFrame pin migration to allow migration from one package to another. The Quartus and MAX+PLUS II software automatically avoids conflicting pins when future migration is set.
(2) Devices in the same package are pin-compatible and have the same number of I/O pins.
(3) Devices in the same package are pin-compatible, although some devices have more I/O pins than others. When planning device migration, use the I/O pins that are common to all devices. The MAX+PLUS II software versions 9.1 and higher provide features to help use only the common pins.
(4) This option will be supported with a 484-pin FineLine BGA package. By using SameFrame pin migration, all FineLine BGA packages are pin-compatible. For example, a board can be designed to support 256-pin, 484-pin, and 672-pin FineLine BGA packages. The Quartus and MAX+PLUS II software automatically avoids conflicting pins when future migration is set.

Table 85 shows the FLEX 10KE device/package combinations that support SameFrame pin-outs for both FLEX 10KA and FLEX 10KE devices. All FineLine BGA packages support SameFrame pin-outs, providing the flexibility to migrate not only from device to device within the same package, but also from one package to another. The I/O count will vary from device-to-device. Therefore, the MAX+PLUS II software versions 9.1 and higher provide features to help designers use only the common pins.

For more information, search for “SameFrame” in MAX+PLUS II Help.

Table 85. FLEX 10K & FLEX 10KE SameFrame Pin-Out Support

Device	256-Pin FineLine BGA	484-Pin FineLine BGA	672-Pin FineLine BGA
EPF10K30E	✓	✓	✓ *(1)*
EPF10K50E EPF10K50S	✓	✓	✓ *(1)*
EPF10K100B	✓		
EPF10K100E	✓	✓	✓ *(1)*
EPF10K130E		✓	✓
EPF10K200E EPF10K200S			✓
EPF10K10A	✓		
EPF10K30A	✓	✓	
EPF10K50V		✓	
EPF10K100A		✓	

Note:

(1) This option will be supported with a 484-pin FineLine BGA package and SameFrame migration.

Notes: